

**VŠB – Technická univerzita Ostrava**  
Fakulta elektrotechniky a informatiky

**BAKALÁŘSKÁ PRÁCE**

**VŠB – Technická univerzita Ostrava**  
**Fakulta elektrotechniky a informatiky**  
**Katedra kybernetiky a biomedicínského inženýrství**

Databáze pro evidenci označených entit sledující koncept Internet of Things

Database for Tagged Entity Storage Following the Internet of Things Concept

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra kybernetiky a biomedicínského inženýrství

## **Zadání bakalářské práce**

Student: **Tomáš Urbanczyk**

Studijní program: B2649 Elektrotechnika

Studijní obor: 3901R039 Biomedicínský technik

Téma: **Databáze pro evidenci označených entit sledující koncept  
Internet of Things  
Database for Tagged Entity Storage Following the Internet of Things  
Concept**

Zásady pro vypracování:

Téma této bakalářské práce sleduje stále aktuálnější koncept Internet Of Things (IOT), který v blízké budoucnosti nalezne své uplatnění v oblasti logistiky, která dnes zasahuje skoro do všech odvětví průmyslu, zdravotnictví i zemědělství.

Předmět, věc nebo i živé zvíře či člověk mohou představovat entitu, jednotku, která je označena jednoduchým a ekonomicky nenáročným identifikátorem, který se nazývá NFC (Near Field Communication) tag, ať už pasivní či aktivní. Tyto tagy vycházejí z technologie RFID (Radio Frequency Identification). Tento identifikátor v sobě nese informaci o stavu dané věci, či člověka. V případě zdravotnictví tento tag může obsahovat informaci o vyšetřeních, které daný pacient absolvoval, nebo léčích, které mu mají být dodávány apod. Další možno aplikací v oblasti medicíny může být evidence zdravotnického materiálu ať už operační náčiní nebo krevní konzervy. V případě logistiky zboží mohou dnes velké obchodní řetězce sledovat stav jednotlivých kusů zboží včetně informací o původu a cestě různými skladišti (čímž je zajištěna ochrana proti krádeži, nebo přetištění data výroby), v případně aktivních tagů také teplotu, v níž byly výrobky uchovány.

Existují mnohé další aplikace pro, které by byl tento systém užitečný. Doplněním o databázi, která představuje elektronický ekvivalent dané entity, která sleduje moderní koncepci IOT (Internet Of Things), se možnost aplikace jenom dále rozšiřuje. Cílem této bakalářské práce je vytvořit systém, který sleduje koncept IOT, a to na zařízeních s OS Android s vestavěnou NFC technologií. Mezi tyto zařízení patří například chytré mobilní telefony smartphone (Google Nexus S) nebo Tablet Sharp-RW-T107.

V této práci bude jako modelové prostředí pro vytvoření databáze použito oddělení urgentního příjmu Fakultní nemocnice v Ostravě Porubě. Úkolem bude vytvoření databáze, která bude schopna evidovat všechny označené entity (pomůcky, léky, nástroje, pacienti, atd.) a uchovávat jejich důležité atributy.

Cíle:

1. Seznámení se se strukturovaným jazykem pro práci s databázemi SQL.
2. Seznámení se s databázovými servery MSSQL, MySQL a PostgreSQL.
3. Seznámení se s prostředím urgentního příjmu FNO a zaevidování všech možných entit, které jsou svým významem a použitím relevantní pro toto oddělení.
4. Vytvoření databáze pro evidenci informací o entitách, které jsou označeny NFC tagy, nebo 2D čárovými kódy, pomocí databázového serveru MySQL.
5. Naplnění databáze testovacím vzorkem nejčastěji používaných entit a otestování maximální možné zátěže databáze (stress test).
6. Zhodnocení dosažených výsledků.

Seznam doporučené odborné literatury:

- [1] MURPHY, Mark L. *Android 2 Průvodce programováním mobilních aplikací*. překlad Jakub MUŽÍK. 1.vyd. Brno:COMPUTER PRESS, 2011. ISBN 978-80-251-3194-7.
- [2] MEIER, Reto. *Professional Android Application Development*. Indianapolis(USA): Wiley Publishing Inc., 2009. ISBN 978-0-470-34471-2. Dostupné také z:  
[http://www.google.cz/url?sa=t&rct=j&q=professional%20android%20application%20development&source=web&cd=2&ved=0CEAQFjAB&url=http%3A%2F%2Fandroidteam.googlecode.com%2Ffiles%2Fprofessional\\_android\\_application\\_development.pdf&ei=G-UFT7DyD8\\_IswaXqumADw&usg=AFQjCNFyzLrx89xFrFq3rvWtSjPohevtQA&sig2=ynMy9esnk8kst9B5kCTcKg](http://www.google.cz/url?sa=t&rct=j&q=professional%20android%20application%20development&source=web&cd=2&ved=0CEAQFjAB&url=http%3A%2F%2Fandroidteam.googlecode.com%2Ffiles%2Fprofessional_android_application_development.pdf&ei=G-UFT7DyD8_IswaXqumADw&usg=AFQjCNFyzLrx89xFrFq3rvWtSjPohevtQA&sig2=ynMy9esnk8kst9B5kCTcKg)
- [3] ECKEL, Bruce. *Myslíme v jazyku Java: knihovna programátora*. Praha:Grada Publishing, 2001. ISBN 80-247-9010-6.
- [4] WIKIPEDIA, the Free Encyclopedia. *NFC, Near Field Communication*. [online] [cit. 2012-10-16]. Dostupné z: [http://en.wikipedia.org/wiki/Near\\_field\\_communication](http://en.wikipedia.org/wiki/Near_field_communication).
- [5] ANDROID DEVELOPERS. The Developer's Guide [online][cit. 2012-10-12]. Dostupné z: <http://developer.android.com/guide/index.html>.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jakub Jirka**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



---

doc. Ing. Jiří Koziorek, Ph.D.  
vedoucí katedry

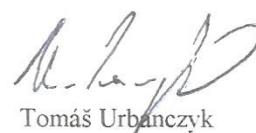
---

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

## **Prohlášení**

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum odevzdání: 7. 5. 2013



Tomáš Urbanczyk

## **Poděkování**

Děkuji Ing. Jakubu Jirkovi, vedoucímu mé bakalářské práce za cenné rady, čas a připomínky, které mi v průběhu psaní poskytnul.

Dále děkuji Bc. Pavlíně Štěpánové, pracovníci urgentního příjmu za výbornou komunikaci a pomoc, které se mi dostalo.

## **Abstrakt**

Tato práce se zabývá tvorbou databáze pro urgentní příjem fakultní nemocnice Ostrava. Databáze sleduje koncept Internet Of Things. Označené prvky se pomocí nízko nákladových identifikátoru uloží do databáze a vytvoří záznam o použití.

Obsahem práce je teoretický úvod k řešení práce. V druhé fázi se zabývám analýzou prostředí urgentního příjmu s obecným popisem pracovního postupu. Dále se zabývám implementování databáze a testováním.

Databáze je tvořena pomocí MySQL databázového systému, který disponuje klientskou aplikací MySQL Workbench 5. 2. 42. CE.

## **Klíčová slova**

databáze, urgentní příjem FNO, internet věcí, MySQL, analýza prostředí

## **Annotation**

This thesis deals with creating the database for emergency department of the faculty hospital of Ostrava. The database is based on the concept of the Internet Of Things. Selected elements are saved to the database using low-cost identifiers that also creates records of use.

The first part of this thesis contains the theoretical analysis that is necessary for the actual work solution. The second part is connected with the working process and working environment of the emergency department. There is also the database implementation and testing's involved.

For the database MySQL database server with client MySQL Workbench 5. 2. 42. CE application was used.

## **Key words**

database, emergency department of FNO, Internet of Things, MySQL, environment analysis

## Seznam použitých zkratek

FNO	Fakultní Nemocnice Ostrava
EAN	European Article Number, Evropský číselný kód
QR	Quick Response, Kód rychlé reakce
RFID	Radio Frequency Identification, Rádio frekvenční komunikace
NFC	Near Field Communication, pole blízké komunikace
SQL	Structured Query Language, Strukturalizovaný jazyk
SŘBD	Systémem Řízení Báze Dat
IO	Integritní Omezení



# Obsah

1	Úvod .....	1
2	Databáze .....	2
2.1	Historie databáze .....	2
2.2	Datové modely .....	3
2.2.1	Lineární datový model .....	3
2.2.2	Hierarchický model dat .....	3
2.2.3	Síťový model dat .....	4
2.2.4	Objektový model dat .....	4
2.3	Relační model dat .....	5
2.3.1	Definice pojmů v relační databázi .....	5
2.3.2	Klíče .....	6
2.3.3	Integritní omezení .....	6
2.3.4	Kardinalita a povinnost členství ve vztahu .....	7
2.3.5	Normalizace .....	7
2.4	Architektura databází .....	8
2.4.1	Centrální architektura .....	8
2.4.2	Architektura file-server .....	9
2.4.3	Architektura klient-server .....	9
2.4.4	Architektura distribuovaných databází .....	10
2.5	SQL Dotazovací jazyk .....	10
2.5.1	Příkazy pro definici dat .....	11
2.5.2	Příkazy pro manipulaci s daty .....	11
3	Databázové systémy .....	13
3.1	MySQL .....	13
3.1.1	Instalace MySQL .....	13
3.1.2	MySQL Workbench CE 5. 2. 42 .....	13
3.1.3	Typy tabulek v MySQL .....	14
3.1.4	Datové typy v MySQL .....	15
3.1.5	Porovnání databázových systémů .....	16
4	Návrh databáze .....	19
4.1	Požadavky na model databáze .....	19
4.2	Konceptuální modelování .....	20
4.2.1	E-R diagram .....	20
4.3	Logický návrh databáze .....	21

4.3.1	Lineární zápis .....	21
4.3.2	Datový slovník .....	21
4.4	Funkční analýza.....	21
4.4.1	Diagram aktivit.....	21
4.4.2	Diagram Datových toků .....	22
4.4.3	Diagram užití .....	23
5	Praktická část bakalářské práce .....	24
6	Analýza urgentního příjmu fakultní nemocnice Ostrava.....	25
6.1	Definování základních pojmů.....	26
6.2	Personál .....	26
6.3	Informační systémy na urgentním příjmu FNO .....	26
6.4	Postup pacienta urgentním příjmem.....	27
6.4.1	Přijetí pacienta .....	28
6.4.2	Uložení pacienta .....	28
6.4.3	Evidence pacienta.....	29
6.4.4	Záznam intenzivní péče.....	29
6.4.5	Smrt .....	29
6.5	Analýza identifikátoru .....	31
6.5.1	Typy identifikátoru .....	31
7	Návrh databáze .....	32
7.1	Konceptuální modelování.....	32
7.1.1	Lineární zápis .....	32
7.1.2	Datový slovník .....	33
7.1.3	E-R diagram .....	35
7.1.4	Popis ER diagramu.....	36
7.1.5	Diagram datových toků .....	36
7.2	Fyzický návrh.....	37
7.2.1	Připojení databáze .....	37
7.2.2	Vytvoření tabulky.....	38
7.2.3	Modelování v MySQL Workbench .....	39
7.3	Test databáze .....	40
8	Zavedení databáze .....	41
8.1	Funkční princip.....	41
8.1.1	Mobilní zařízení.....	42
8.2	Přínos databáze.....	42
9	Závěr.....	43

# 1 Úvod

Dnešní moderní doba se snaží pomocí počítačů strukturalizovat svět kolem sebe. Strukturalizovat se rozumí data uvádět jako informace v efektivní a jedinečné formě. Takto uskládaná data se dále dají zpracovávat a analyzovat.

Díky právě se rozvíjejícímu trendu, který sleduje koncept Internet Of Things (IOT) můžeme do jednotlivých, ať už zdravotnických nebo jiných oborů vnést onu strukturalizaci. Představme si možnost absolutně kontrolovat dění na pracovišti, mít okamžitý přehled o stavu označeného prvku s kteréhokoliv počítače nebo i tzv. chytrého mobilního telefonu<sup>1</sup>. Takovou možnost nám umožňuje moderní technologie s využitím Radio Frekuensi Identification (RFID). Pomocí nízkonákladových identifikátorů zvaných jako Near Field Communication (NFC) vycházejících právě z technologie RFID označujeme jakoukoliv věc, předmět nebo i živou bytost.

Základním kamenem této bakalářské práce je vytvořit databázi pro urgentní příjem fakultní nemocnice Ostrava (dále jen FNO). FNO se snaží zefektivnit péči o pacienta a získat strukturalizovaný zápis práce k pacientovi. K tomuto projektu se dále bude rozvíjet i wifi kontrola pozice pacienta. Tyto projekty budou dohromady tvořit funkční celek, který nejenže urychlí dobu pacienta strávenou na urgentním příjmu, ale také by měl pomoci k organizaci práce na urgentním příjmu. Databáze bude evidovat jednotlivé pacienty a k nim jednotlivé výkony prováděné na pacientovi (např. podání léku, vyšetření, výkony a další). Evidence bude fungovat tak, že do databáze se budou ukládat identifikační data označených prvků (např. léku, přístrojů), které se na pacienta použili za pomoci mobilního telefonu<sup>2</sup>. Tímto se vytvoří elektronický záznam intenzivní péče a zpřístupní rychlý přehled o pacientovi. Je zapotřebí si uvědomit, že ne všechny údaje se dají označit identifikátory.

Obsahem této práce je analyzovat prostředí urgentního příjmu FNO a jednoznačně určit prvky, které se mohou evidovat mobilním telefonem. Prvky, které se musí označit a prvky, které označit nejdou.

---

<sup>1</sup>Za předpokladu dostupné internetové sítě

<sup>2</sup>Mobilní telefon s aplikací schopnou číst identifikační kódy

## 2 Databáze

Databáze (neboli datová základna) je z českého slova smyslu složeno ze slov data a báze.

**Data** můžeme chápat, jako souhrn naměřených nebo zaznamenávaných údajů z reálné skutečnosti.

*„V praxi je datum běžně přisuzován význam zpráv.“ (Koch, Neuwirth, 2008, s. 5). [1]*

**Báze** (základna) má v češtině mnoho významů. V informační technice (IT) označuje množinu utříděných hodnot.

V literatuře se databáze definuje takto: *„Databáze je kolekce vzájemně souvisejících datových položek, které jsou spravovány jako jediná jednotka.“ (Oppel, 2008, s. 9). [2]* Nebo také takto: *„Množinu datových souborů, uchovávajících data o nějakém uceleném úseku reality, nazýváme databází.“ (Šarmanová, 2007, s. 10). [3]*

### 2.1 Historie databáze

Historie databáze sahá na počátek civilizace. Přibližně do doby, kdy člověk dokázal své myšlenky nějakým způsobem archivovat. Vznik papíru měl pro lidstvo gigantický rozměr. Použitím papíru vznikali první kartotéky takové, jaké si ještě pamatujeme např. z nemocnic. Kartotéky splňovaly požadavky databáze, ale při větším množství záznamu nebo vlivem selhání lidského faktoru nebyli dostatečně efektivní. V roce 1890 se v Americe konal historicky první součet lidu, kde se ke zpracování výsledku použil stroj. Rozvoj počítačů v 50 letech 20. století dopomohl k elektronickému ukládání dat. Tyto data však byla jen surová a jakákoliv operace s nimi byla pomalá. A tak v roce 1965 na konferenci CODASTYL byla zvolena skupina pro vývoj koncepce databázových systémů. Kolem roku 1970 začaly vznikat první síťové, ale i hierarchické databáze se systémem řízení báze dat (SŘBD).

*„Programový systém umožňující definování datových struktur a datových souborů, řešící fyzické uložení dat ve vnější paměti počítače, umožňující manipulaci s daty a formátování vstupních i výstupních informací, nazýváme **systémem řízení báze dat**.“ (Šarmanová, 2007, s. 10). [3]*

V roce 1970 vznikají také relační databáze. Relační databáze jsou vlastně tabulky, což s příchodem dotazovacího jazyka Structured Query Language (SQL) v roce 1974, vedlo k zatím spolu s objektově – relační technologií nejpopulárnějšími datovými modely. [6]

## 2.2 Datové modely

Z hlediska způsobu ukládání dat a vazeb mezi nimi dělíme databáze do pěti základních logických typů:

- Lineární
- Síťový
- Hierarchické
- Relační
- Objektové

### 2.2.1 Lineární datový model

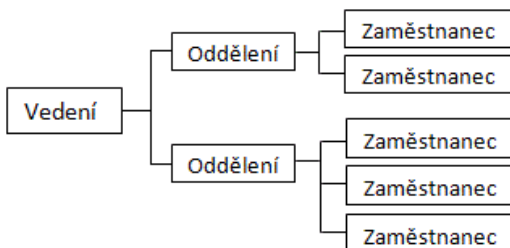
Tento datový model je sestaven s posloupností dat. Data nemají na sebe žádnou vazbu, kromě dat před a za sebou. Jsou set řízený podle jednoho indexu vyhledávání. Představují ten nejzákladnější model a reprezentují ho např. kartotéky.



Obrázek 1: Lineárního datového modelu  
(Vlastní zpracování)

### 2.2.2 Hierarchický model dat

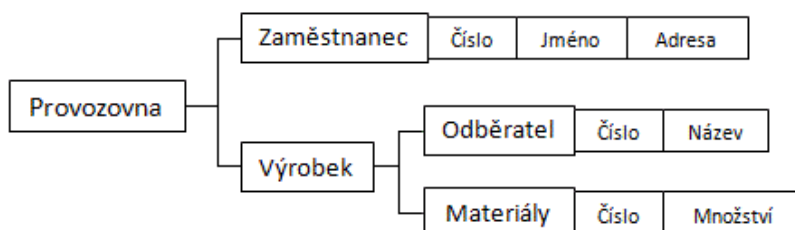
Data jsou seřazena do stromové struktury. Stromová struktura je charakteristická svým uspořádáním. Každý záznam představuje jakési rozvětvení, kdy další záznam je potomek záznamu předchozího. Vztahy mezi jednotlivými daty jsou mezi rodičem a potomkem, tedy nahoru a dolů, a také mezi potomky do stran. Největšími nevýhodami hierarchického uspořádání je složitá operace vkládání a rušení záznamů a v některých případech i nepřírozená organizace dat. [4],[5]



Obrázek 2: Hierarchický model dat (Dostupný z: <http://www.databaze.chytrak.cz/modely.htm>)

### 2.2.3 Síťový model dat

Síťový model dat je v podstatě zobecněním hierarchického modelu, který doplňuje o mnohonásobné vztahy (sety). Tyto sety propojují záznamy různého či stejného typu, přičemž spojení může být realizováno na jeden nebo více záznamů. Přístup k propojeným záznamům je přímý bez dalšího vyhledávání, k dispozici jsou operace: nalezení záznamu podle klíče, posun na prvního potomka v dílčím setu, posun stranou na dalšího potomka v setu, posun nahoru z potomka na jeho rodiče v jiném setu. Nevýhodou síťové databáze je zejména nepružnost a obtížná změna její struktury. [4],[5]



Obrázek 3: Síťový model dat (Dostupný z: <http://www.databaze.chytrak.cz/modely.htm>)

### 2.2.4 Objektový model dat

Objektový model dat v databázových systémech vychází ze známých principů objektově orientovaného modelování a programování. Je však dále obohacen o techniky perzistence, reprezentace vztahů, dotazování, transakčního přístupu apod.

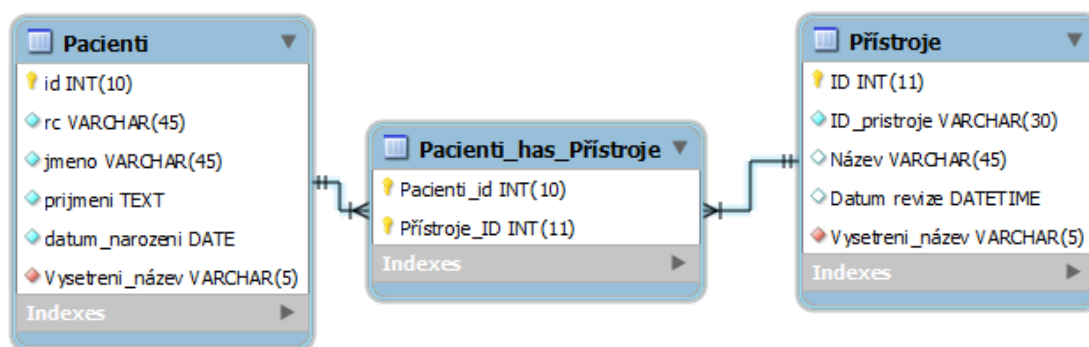
Tento model je velmi podobný síťovému modelu, ale dokáže zapisovat objekty lépe a složitou koncepcí data lépe ovládat. Mezi objekty existuje skládání, dědění, závislost, klasifikace podle tříd. Strukturované informace není třeba rozdělovat jako v Relačním Databázovém Modelu (RDM) do tabulek. [4],[5]



Obrázek 4: Objektový model dat

## 2.3 Relační model dat

Relační databázový model je z uvedených nejmladší a zároveň nejpoužívanější. I v této práci byl použit relační datový model, a tak na něj byl kladen větší informační důraz. V roce 1970 byl popsán Dr. Coddem. V současnosti je nejčastěji využíván u komerčních SŘBD. Model má jednoduchou strukturu, data jsou organizována v dvourozměrných tabulkách, které se skládají z řádků a sloupců. V těchto tabulkách jsou prováděny všechny databázové operace. Velkou předností relačního modelu je schopnost tabulky spojovat a vytvářet další dvourozměrné tabulky. [4],[5]



Obrázek 5: Relační model dat (Vlastní zpracování)

### 2.3.1 Definice pojmů v relační databázi

#### Typ entity

Je známo, že relační databáze shromažďuje data do tabulek. Tabulky jsou sestaveny tak, aby setřídily podle charakteru dat. To znamená, že tabulka s typem entity pacient bude shromažďovat záznamy všech pacientů.

#### Entita

Entita je osoba, místo, věc, událost nebo myšlenka, o níž shromažďujeme nějaká data. Jinými slovy entity jsou „předměty“ z reálného světa, které jsou pro nás dostatečně zajímavé, že o nich sledujeme údaje a zaznamenáváme je do databáze. [7]

V databázi přesněji pak v typech entit (tabulkách) je entita reprezentovaná řádkem. Například máme evidovat pacienty. Každý záznam pacienta je entita.

#### Atribut

Atribut je vlastnost entity. Popisuje důležité informace entity tak, aby možno co nejvíc přiblížil problematiku reálného světa. Čím více bude entita popsána, tím více budeme shromažďovat dat a systém bude chytřejší na úkor množství evidovaných dat.

V databázi přesněji pak v typech entit (tabulkách) je atribut reprezentovaný sloupcem. **Složený atribut** – skupina atributů, která má společný význam nebo použití. Typickým příkladem je atribut adresa, který je složen z jednoduchých atributů PSČ, město, ulice, číslo domu.

### Doména

Množina přípustných hodnot přiřazená jednomu nebo více atributů. Například doména všech hodnot čísla \_pracovníka.

### Relace

Relace jsou informační mosty tabulek. Relací propojíme tabulky a tím vytvoříme nad tabulku, která bude rozšířena o informace právě propojených tabulek. Např. tabulku pacienti, kterou propojíme s tabulkou přístroje. Dovedeme si tak vyhledat, jaký přístroj byl použit na pacienta.

## **2.3.2 Klíče**

Existují speciální druhy atributů, které umí jednoznačně určit pozici nebo entitu v tabulce.

Superklíč - Atribut, nebo množina atributů, které jednoznačně identifikují entitu v relaci.

Kandidátní klíč - Superklíč, který obsahuje minimální počet atributů potřebných k jednoznačné identifikaci entity. Je zapotřebí vždy definovat jako NOT NULL.

Primární klíč - Jeden vybraný kandidátní klíč

Alternativní klíč - Zbývající kandidátní klíče.

Cizí klíč - Představuje odkaz mezi tabulkami. Nejčastěji to bývá primární klíč vázané tabulky relací.

## **2.3.3 Integritní omezení**

Integritní omezení v dané databázi vyplývá z omezení reálného světa. Zabraňuje chybnému vyplnění dat. Proto, aby databáze fungovala správně je zapotřebí dodržovat obecná ale také i specifická pravidla relační databáze. Rozlišujeme integritní omezení:

- **Pro entity** - v tabulce se nesmějí vyskytovat dva totožné záznamy (primární klíč).
- **Pro atributy** – atribut je vždy omezen svým datovým typem. Tabulka nesmí obsahovat nedefinované buňky.
- **Referenční integrita** – tabulka podřazená tabulkou nadřazenou musí obsahovat v cizím klíči stejný záznam primárního klíče tabulky nadřazené nebo hodnotu NULL.
- **Vazební integrita** – kardinalita (vysvětleno 2.3.4)



NOT NULL - Zaručí, že každý atribut označený NOT NULL bude vždy vyplněn. Předjedeme, tak nekompletním informacím o objektu. Pomáhá k vyššímu výkonu databáze.

### 2.3.4 Kardinalita a povinnost členství ve vztahu

#### Kardinalita

Je logické omezení vztahu mezi tabulkami na poměry 1:1, 1:N, M:1 a M:N. Pro určení kardinality vztahu používáme větu, kdy se ptáme, např. jeden pacient může mít jedno vyšetření nebo N vyšetření může mít N pacientů. Dostaneme vazbu M:N a musíme zakročit vazební tabulkou. **Vazební tabulka** je nová tabulka, která musí obsahovat cizí klíče tabulek a upřesní tak vazbu na 1:N a N:1.

1:1 – Vztah, ve kterém na obou stranách vystupuje pouze jeden objekt dané entity (např. jeden host má jednu židličku, jedna židlička má jednoho hosta)

1:N – Na jedné straně je jediný objekt, který je ve vztahu s jedním nebo více objekty na straně druhé (např. jeden žák má jednu studijní skupinu. Jedna studijní skupina má N žáků.).

M:N – Vztahy, kde vystupuje více objektů na obou stranách vazby (např. pacient má n přístrojů jeden přístroj může mít M pacientů)

#### Povinnost členství ve vztahu

Povinnost členství je nutnost objektu patřit objektu druhému. Je to vztah, který musí být dodržený anebo právě nemusí. Také může být na jednom objektu povinný a v relaci na druhém objektu nepovinný. Je dobré to vědět. Např. pacient musí být vyšetřen, ale nemusí dostat lék, ale každý použitý lék musí mít pacienta.

### 2.3.5 Normalizace

*„Normalizace je proces, s jehož pomocí lze databázi zbavit strukturálních vad. Normalizace je souhrnem několika normálních forem, což jsou množiny pravidel udávajících, jaká by měla a neměla být struktura našich tabulek. Normalizační proces spočívá v rozdělování tabulek do menších tabulek, které povedou k lepšímu návrhu databáze.“<sup>3</sup> [8]* Vždy platí, že vyšší norma musí splňovat normu nižší!

#### První normální forma

Základní normální forma, která upozorňuje na základní dodržení atomických atributů. Atomických dále nedělitelných atributů. To znamená, že atribut musí obsahovat jedinou hodnotu. Např. adresa je neatomický dále dělitelný atribut na města ulice obce. [8]

---

<sup>3</sup> LUKE Welling a Laura THOMSON, 2005, str. 46

### Druhá normální forma

“Relace je ve druhé normální formě, pokud je v první normální formě a navíc všechny její atributy jsou závislé na celém kandidátním (primárním) klíči.“ [1] Druhá norma vyžaduje, aby se každý sloupec podílel na funkčním popisu entity a byl závislý na primárním klíči. Pokud primární klíč jasně nezahrne množinu atributů, pak model tabulky nepatří do druhé normální formy. Musíme tabulku rozdělit a jednoznačně popsat.

### Třetí normální forma

Jestliže druhá normální forma hovoří o závislosti atributů na primárním klíči, poté třetí normální forma hovoří o dominanci primárního klíče v tabulce. Žádný atribut nesmí být závislý na ničem jiném než na primárním klíči. Schéma nesmí obsahovat žádné zprostředkované závislosti. Tj. závislost atributů na jiném klíči než primárním. [8]

### Boyce - Coddova normální forma

BCNF platí, pokud jsou dodrženy formy předešlé a vyhovuje formě první. Podmínkou je, že všechny funkční závislosti musí mít superklíč na levé straně a mezi kardinálními klíči není žádná funkční závislost. [8]

### Vyšší normální forma

Existují vyšší normální formy (čtvrtá, pátá atd.), ale ty mají význam spíše pro akademickou práci, než pro praktický návrh databáze. Forma 3NF (nebo BCNF) je dostačující k tomu, abyste se vyhnuli problémům s redundancí dat, na něž narazíte.[8]

Vyšší norma platí, pokud jsou splněny podmínky nižších forem a pokud tabulka popisuje jen jeden fakt nebo souvislost.

## **2.4 Architektura databází**

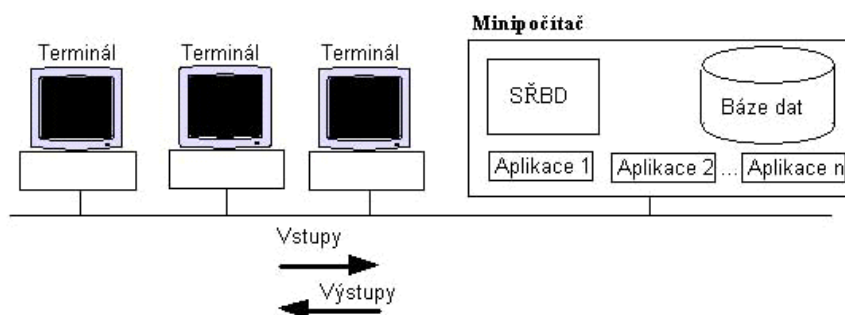
Pod tímto pojmem se rozumí jednotlivý přístup k datům. K datům se přistupuje podle potřeby. Je několik možností přístupu:

- Centrální architektura
- Architektura file-server
- Architektura klient-server
- Distribuovaná databáze

### **2.4.1 Centrální architektura**

Velmi vhodné použití centrální architektury se využívá při terminálech. Princip spočívá v tom, že existuje mnoho přístupových míst, která jsou však jen infomační a editační místa. Dotazy

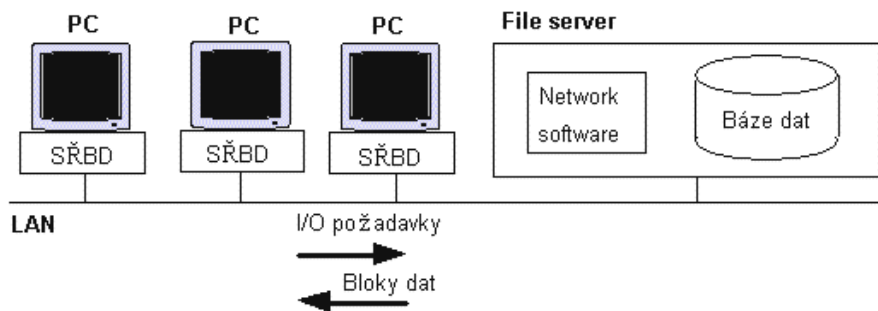
pak směřují do určitých aplikací centrálního počítače, kde jsou zpracovány SŘBD a použity na data. Výstup z centrálního počítače se pak zobrazí na konkrétním terminálu. Musíme však brát v úvahu, že je mnoho dotazovacích míst, tudíž na centrálním počítači vznikají kolony, které se projevují určitým zpožděním vlivem zatížení. [4],[5]



Obrázek 6: Centrální architektura

#### 2.4.2 Architektura file-server

Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jednomu datu, musí SŘBD používat vhodný systém zamykání (položek nebo celých tabulek). [4]

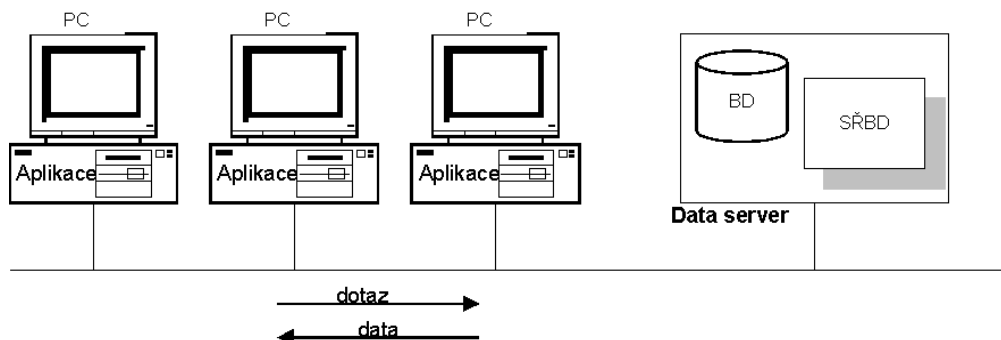


Obrázek 7: Architektura file-server

#### 2.4.3 Architektura klient-server

V práci byla použita právě tato architektura databáze, která komunikuje se školním serverem. Je to způsob řízení dat přes nainstalovanou aplikaci, která zprostředkovává ovládání další aplikace nad uloženými daty na serveru. Server není přímo spojen s klientem, ale zprostředkování serveru klientovi se provádí přes síťové připojení. Tedy klientská část je nainstalována na

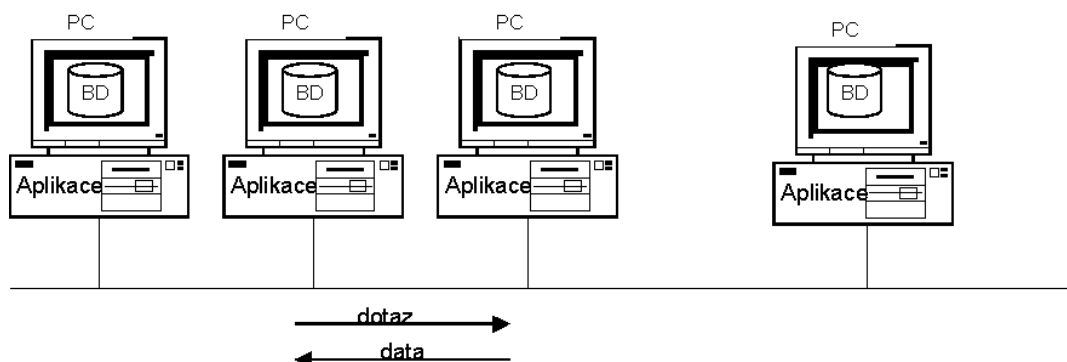
uživatelském počítači a server může a nemusí být odděleně. Takto funguje většina webových stránek a dalších aplikací. [4]



Obrázek 8: Architektura klient-server (Dostupné z: [http://homen.vsb.cz/~s1i95/ISVDAS/IS/IS\\_db\\_sys.htm](http://homen.vsb.cz/~s1i95/ISVDAS/IS/IS_db_sys.htm))

#### 2.4.4 Architektura distribuovaných databází

Tato architektura se navenek tváří jako jednotný celek, říkáme *transparentnost*. Ovšem server je rozčleněn do uživatelských míst a řízen přes SŘBD. Na každém uživatelském počítači je nainstalována aplikace společně se serverem a z části uložených dat. Pro zbytek dat je zapotřebí vyniknout do dalších uživatelských počítačů.



Obrázek 9: Architektura distribuovaných databází (Dostupné z: [http://homen.vsb.cz/~s1i95/ISVDAS/IS/IS\\_db\\_sys.htm](http://homen.vsb.cz/~s1i95/ISVDAS/IS/IS_db_sys.htm))

### 2.5 SQL Dotazovací jazyk

Typ neprocedurálního databázového jazyka založený na teorii množin a relační algebře optimalizovaný pro práci s relačními databázemi. Je využitelný pro všechny typy operací s daty: založení databáze, definování tabulek a indexů, stanovení integritních omezení, vkládání a změny dat, výběr dat z databáze, zobrazení dat. Slovník je odvozen z angličtiny, jednoduchá syntaxe připomíná věty přirozeného jazyka. [9]

"SELECT Kola FROM silnicek WHERE Cena>30 000"

Vyber kola z kategorie silniček, jejichž cena přesahuje 30 000 Kč". V současné době se používá ve většině komerčně dostupných databázových systémů, buď k interaktivní práci s daty nebo jako komunikační nástroj v databázích klient/server. [9]

### 2.5.1 Příkazy pro definici dat

Příkazy nebo-li jazyk pro definici dat, umožňuje definici dat. Pomocí tohoto jazyka můžeme měnit strukturu databáze. Díky databázovým systémům se tyto dotazy k uživateli prakticky nedostanou. Nic méně je dobré znát množinu základních příkazů:

- CREATE DATABASE - vytvoření nové databáze
- CREATE TABLE - vytvoření tabulky
- CREATE VIEW - vytvoření pohledu
- DROP TABLE - zrušení (smazání) tabulky
- DROP VIEW - zrušení pohledu
- ALTER TABLE - úprava (modifikace) dat tabulky
- ALTER VIEW - úprava nastavení pohledu

### 2.5.2 Příkazy pro manipulaci s daty

Jazykem pro manipulaci s daty ovládáme a manipulujeme s daty, které již mají definovanou strukturu. Data například vkládáme, vyhledáváme, aktualizujeme nebo také mažeme. Základní nejpoužívanější množina příkazů:

- SELECT - výběr dat
- INSERT - vložení dat
- DELETE - smazání dat
- UPDATE - aktualizace dat

Jak už bylo zmíněno výše, uživatel se v nových verzích databázových systému, málo kdy setká s jazykem SQL. Ale pro úplnost je dobré uvést další nezbytné příkazy. Mezi ně patří i možnost vytvářet, měnit, mazat a přidělovat privilegia uživatele:

- CREATE USER - vytvoření uživatele
- ALTER USER - změna konkrétního uživatele
- DROP USER - odstranění uživatele
- GRANT - přidělení privilegia

Příkazy, kterými lze provádět transakce:

- START TRANSACTION zahajuje provádění transakce
- SET TRANSACTION - nastavuje vlastnosti právě probíhající transakce
- COMMIT - potvrzuje a ukončuje transakci
- ROLLBACK - zruší zahájenou transakci

Agregační funkce:

- COUNT(\*) - počet záznamů včetně těch, které obsahují NULL hodnoty
- COUNT(*položka*) - počet záznamů, neuvažují se položky s hodnotou NULL
- AVG (*výraz*) - aritmetický průměr
- SUM (*výraz*) - součet
- MIN (*výraz*) - minimální hodnota
- MAX (*výraz*) - maximální hodnota
- STDEV (*výraz*) - směrodatná odchylka základního souboru číselných výrazů
- STDEVP (*výraz*) - odhad směrodatné odchylky základního souboru ze směrodatných odchylek skupin vytvořených pomocí GROUP BY
- VAR (*výraz*) - rozptyl základního souboru číselných výrazů
- VARP (*výraz*) - odhad rozptylu základního souboru z rozptylů skupin vytvořených pomocí GROUP BY

### **3 Databázové systémy**

V bakalářské práci byl použit systém s relačním modelem dat. Databázových systému podporujících relační model je celá řada. Informační důraz kladu na MySQL, jelikož s tímto systémem byla tvořena práce. Všechny zmíněné systémy fungují s dotazovacím jazykem SQL.

Databázový systém se skládá z programu pro práci s databázemi a celé řady podpůrných programů. Databázový systém obstarává přístup k datům. Pomocí nich můžeme vkládat, editovat, mazat, vytvářet, spojovat a mnoho dalšího. Dokonce systém můžeme zpřístupnit jako aplikaci, kdy uživatel vlastně neví, že pracuje s databázovým systémem.

#### **3.1 MySQL**

MySQL je databázový systém vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, dceřinou společností je Oracle Corporation. Je dostupný jako open source, ale také pod placenou licenci. Považuje se za nejpoužívanější relační databázový systém, který vyčnívá svou stabilitou a poměrně slušnou rychlostí. Databáze MySQL je jeden z prvních hojně rozšířených systémů. Práce s tímto systémem se dá využít v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo .NET. Momentální nejnovější verze MySQL, kterou Oracle nabízí je 6. 5. 11. V bakalářské práci byla použita verze Workbench 5. 2. 42 CE. [6],

##### **3.1.1 Instalace MySQL**

MySQL je zdarma dostupný z oficiálních stránek výrobce [www.mysql.org](http://www.mysql.org). MySQL, je multiplatformní systém, který lze nainstalovat například na Linux, Microsoft a další. K dispozici je instalační průvodce, který dohlédne na správnou instalaci systémů, která není nikterak jednoduchá. Dostaneme na výběr instalaci klienta, serveru anebo kombinaci klient-server. Po instalaci je dobré zadat heslo a doporučuje se i vyplnit root (kořen) pro bezpečné používání databáze. Velmi oblíbený pro programátory webových stránek je balíček Linux, PHP, MySQL.

##### **3.1.2 MySQL Workbench CE 5. 2. 42**

MySQL disponuje mnoho podpůrných softwaru pro usnadnění práce programátora. Jedním z takových softwarů je MySQL Workbench. Je to grafické uživatelské rozhraní umožňující správu serveru a dokonce i konceptuální modelování. Prostřednictvím tohoto se odstartovala pětiková řada MySQL. Uživatelská část, kdy uživatel vyplňuje, vytváří, propojuje a všechno ostatní aniž by věděl, že pracuje s jazykem SQL. Uživateli to připomene při přeložení, kdy se mu ukáže hotový zápis SQL.

### 3.1.3 Typy tabulek v MySQL

Pracovní verze MySQL nabízí 20 typů úložných enginů (storage engine), které se liší svými možnostmi, použitím a způsobem ukládání dat do souborů. Vyjmenuji a popíši jen ty nejzákladnější. [6]

#### InnoDB

- Podporují transakce.
- Podporují uzamykání na úrovni řádku.
- Velikost tabulek není omezena, protože jej lze ukládat do více než jednoho souboru.
- Poskytují konzistentní neuzamčené čtení v dotazu SELECT.
- Jsou přenositelné z jednoho systému na druhý.
- Zabírají více místa na disku než tabulky MyISAM.
- Mezi tabulkami InnoDB jsou podporovány cizí klíče.[8]

#### MyISAM

- MyISAM je výchozí typ tabulek. Nepodporují transakce, ale jsou velmi rychlé.
- Jsou přenosné z jednoho systému na druhý.
- Jejich velikost je omezena operačním systémem.
- Umožňují kompresi tabulek.
- Jedna tabulka může mít maximálně 64 klíčů. [8]

#### BDB

- Podporují transakce.
- Podporují uzamykání na úrovni tabulek.
- Nejsou přenosné. [8]

#### MERGE

- Díky tabulkám MERGE lze zpracovávat více tabulek MySQL jako jedinou tabulku, čímž lze obejít omezení velikosti tabulek MyISAM. [8]

#### HEAP

- Ukládání dat pouze do paměti, v případě výpadku proudu budou ztracena.
- Tabulky HEAP jsou výjimečně rychlé, pokud je k dispozici dostatek fyzické paměti.
- Nepodporují AUTO\_INCREMENT, TEXT nebo BLOB. [8]



### 3.1.4 Datové typy v MySQL

Datové typy umožňují charakterizovat a formovat data. Každý atribut z reálného prostředí musí být definován na vypovídající hodnotu. S datovými typy se setkáte vždy, když se rozhodnete vytvářet tabulku. Nejzákladnější datové typy:

- Číselný typ
- Řetězec
- Datum
- Speciální datové typy

#### Číselný datový typ

Pro uložení čísel používáme číselné datové typy. Liší se v maximálním rozsahu uložení, ale také v typu čísla. Čísla máme celá, s desetinou čárkou, se znaménkem anebo čísla, která se zaokrouhlují.

**Tab. 1: Tabulka číselných datových typů v MySQL**

Datový typ	Rozsah	Místo
TINYINT	od -128 do +127	1 byte
SMALLINT	od -32 768 do 32 767	2 byte
MEDIUMINT	od -8 388 608 do +8 388 607	3 byte
INT	od -2 147 483 648 do +2 147 483 647	4 byte
BIGINT	od -9 223 372 036 854 775 808 do +9 223 372 036 854 775 807	8 byte
BIT	true, falls, NULL	1 bit
FLOAT	od -3.402823466E+38 do 3.402823466E+38	4-8 byte
REAL	od -1.7976931348623157E+308 do 1.7976931348623157E+308	4
DECIMAL (m,d)	"m" (počet číslic celkem) "d" (počet desetinných míst)	5-17 byte
NUMERIC (m,d)	stejný pro DECIMAL	5-17 byte

#### Řetězcové datové typy

MySQL umožňuje ukládání různých řetězcových a textových typů. Znaky, které ukládáme do atributů mají převážně odlišnou hodnotu znaku. Proto, abychom nezpomalovali databázi, jsou k dispozici pole, jak s pevnou délkou, kde chybějící data systém doplní mezerami, tak data s proměnlivou délkou.

**Tab. 2: Řetězcové datové typy v MySQL**

Datový typ	Délka pole	Poznámka
CHAR (n)	Od 0 do 255	1 byte na znak, 8000 znaků
VARCHAR (n)	Od 0 do 255	1 byte na znak, 8000 znaků
TEXT	Od 0 do 65 535	1 byte na znak,
BLOB	Od 0 do 65 535	Ukládá binární data
ENUM ('m','n')	Od 0 do 65 535	Množina prvků sniž 1 platí
SET	64 prvků	Množina prvků sniž n platí
LONGTEXT	Do 4 294 967 295	-

Datum a čas

MySQL používá, pro uložení času a data, různé datové typy. V této bakalářské práci byl použit zejména DATETIME. Tento typ vyhovoval především možnosti uložení data s časem.

**Tab. 3: Časové a datové typy v MySQL**

Datový typ	Rozsah	Přesnost	Místo
DATE	od 0001-01-01 do 9999-12-31	1 den	3 byte
YEAR	year (2) 99 let	1 den	3 byte
DATETIME	od 1000-01-01 00:00:00 do 9999-12-31 23:59:59	0,00333s	8 byte
TIMESTAMP	od 1970-01-01 00:00:00 do 2037-01-01 00:00:00	0,00333s	8 byte
TIME	od 00:00:00.0000000 do 23:59:59.9999999	100ns	4-6 byte

Ostatní datové typy

Ostatní datové typy jsou již velmi unikátní a používají se zřídka jako datový typ IMAGE, který uloží obrázek do velikosti 2GB, ale velmi častý datový typ je NULL. Tento datový typ má nejmenší velikost ze všech, proto se používá všude tam, kde jsou prázdná pole, dále to jsou datové typy XAML, GEOMETRY atd.

**3.1.5 Porovnání databázových systémů**

Porovnávat databázové servery je velmi těžké, jelikož každý výrobce věří, že ten jeho systém je nejlepší, nejrychlejší, nejvýkonnější a samozřejmě nejbezpečnější. Nejlepším způsobem, jak zjistit po jakém databázovém systému šáhnout, je si jej vyzkoušet. Do porovnání systému byly zahrnuty tyto databázové systémy:

- MySQL
- PostgreSQL
- Microsoft SQL Server

### Dostupnost

**MySQL** je dostupný volně jako open source s GPL licenci, ale také s komerční placenou licenci. Multiplatformní záležitost.

**PostgreSQL** je šířen pod [BSD](#) licenci, která je nejliberálnější ze všech open source licencí. Tato licence umožňuje neomezené bezplatné používání, modifikaci a distribuci PostgreSQL a to ať pro komerční nebo nekomerční využití. PostgreSQL můžete šířit se zdrojovými kódy nebo bez nich zdarma nebo komerčně. [6] Původně vytvořen na Linuxové platformy. Dnes lze použít prakticky na vše.

**MMSQL** jako jediný systém je close source zdarma dostupný pouze jako trials verze. V současné době je volně dostupný systém Microsoft SQL server 2012, který však má jisté omezení. Firma Microsoft se snažila zaplnit místo v databázových systémech, a tak přišel na trh server SQL 1.0 primárně vytvořený pro Microsoft Windows.

### Výkon

Momentálně se tvrdí, že nejvýkonnější a nejrychlejší databáze na trhu je MySQL. Má silné postavení, a také byl pro rychlost navržený. Ovšem, že na úkor funkcí. Jediný systém, který může konkurovat je PgSQL. PostgreSQL disponuje mnoha funkcemi, jelikož je vyvíjena stovkami programátorů po celém světě. MySQL již ale nezaostává a s každou novou verzí přináší potřebné funkce jako transakce, pohledy, trigger a mnoho dalšího. MMSQL výkonově malinko zaostává a je paradoxní, že volně dostupné databázové systémy dokáží držet krok s komerčními systémy. Při studii databázových systému byl zjištěn moderní databázový Firebird, který si ve výkonostních testech vedl velmi dobře.

### Použití

**MySQL** získal velkou podporu od webhostingových společností, kdy jej nabízí jako součást svého portfolia svých služeb. Velmi často je využíván společně s Linux, PHP a Apache jako základní software webového serveru, ale je také vhodný pro středně a velké databáze.

**PostgresSQL** je jedna z největší škálou funkcí, která se stává databázovým standardem. Mnohdy se může stát, že narazíme na databázi, která bude obsahovat celky právě z PostgreSQL jako např. Vertica. Velmi vhodný pro středně velké až objemné databáze. PostgreSQL má velmi rozsáhlou dokumentaci v několika jazycích.

*„Microsoft SQL Server je relační databázový a analytický systém pro e-obchody, byznys a řešení datových skladů. Pomáhá snižovat celkové náklady na vlastnictví a zvyšovat efektivitu vaší*

*organizace prostřednictvím správy podnikových dat a práce s obchodními informacemi (business intelligence).* “[6]

#### Vlastní závěr

Je těžké se orientovat ve světě databázových systémů, jelikož vývoj v této oblasti jde velmi rychle. Volba databázového systému je na samotném uživateli, který má jistě na výběr.

Na vrcholu databázových systému se podle mého názoru bude držet MySQL hned za ní PostgreSQL. Microsoft SQL Server podle mě je trochu neutrální databázový systém. Osobně si myslím, že v nejbližší době se databáze budou směřovat na hybridní objektově relační systémy. Velmi lukrativní databázový systém, který jsem zde neuváděl, ale zmínil je Firebird, který má velké ambice.

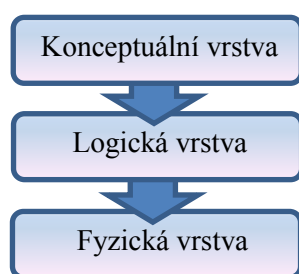
## 4 Návrh databáze

Návrh databáze je spojen s návrhem informačního systému. V této bakalářské práci je však úkolem vytvořit pouze databázi a analyzovat prostředí. Při návrhu databáze je vycházeno z vodopádového modelu.

1. Specifikace požadavků
2. Návrh
3. Implementace
4. Integrace
5. Testování a ladění (validace)
6. Instalace
7. Údržba

První dva body modelu se vztahují k modelu obchodních požadavků. Fáze návrhu, kdy zadavatel zadává své požadavky na funkčnost systému. Je zapotřebí vymezit základní hranice systému a vytvořit primární návrh ve spolupráci se zadavatelem.

Zbylé body můžeme zahrnout do modelování databáze. Modelování databáze rozdělujeme na dvě části logické a fyzické, přičemž se na to můžeme dívat ve třech vrstvách.



### 4.1 Požadavky na model databáze

Požadavky nejsou hlídané, a tak je zapotřebí si stanovit základní pravidla pro tvorbu databázového modelu. Je dobré znát alespoň tyto požadavky:

- Přehlednost uložených dat
- Zabezpečení
- Stabilita (současné připojení více uživatelů, platnost dat, předcházení technických problémů)
- Rychlé a přehledné zadávání dotazů, vyhledávání, mazání a přidávání dat
- Zabránění redundance (duplicita dat)
- Vykonatelný

## 4.2 Konceptuální modelování

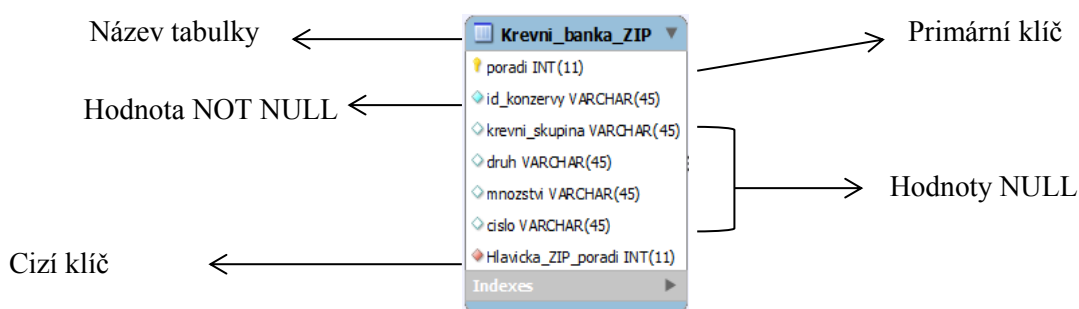
Konceptuální nebo-li datové modelování je základním krokem k popisu reality. Konceptuální návrh určuje, co je obsahem systému. Do tohoto modelování spadají analýzy prostředí komunikace se zadavatelem a nashromáždění potřebných informací k vytvoření účelné aplikace. Velmi vhodný způsob vyjádření základní vztahů je objektově relační model dat.

### 4.2.1 E-R diagram

Objektově relační diagram (Diagramy Entity Relationship), také známý jako ER diagram. Tento ER diagram poskytne programátorům databáze cenný nástroj pro modelování vztahů mezi databázovými subjekty v jasném a přesném formátu. Tento diagram používá standardní řadu blokových tvarů a spojů k popisu struktury databáze srozumitelný pro všechny databázové profesionály. Mnoho balíčků databázových systémů, včetně aplikace Microsoft Access, SQL Server a Oracle poskytuje automatizované metody pro rychlé vytvoření ER diagramu z existujících databází.

#### Objekt

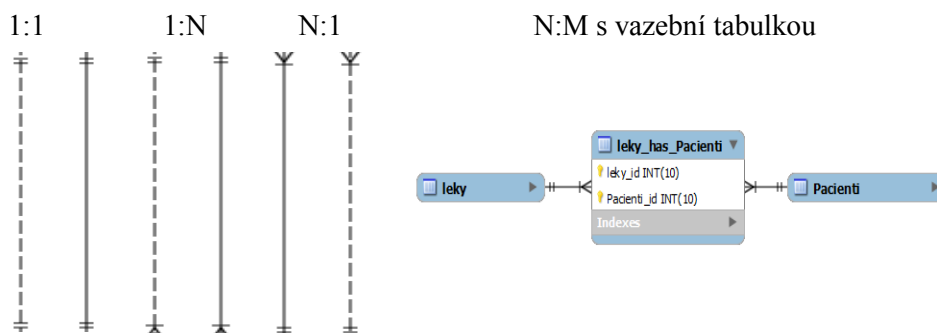
Objekt nebo-li typ entity, tabulka. Pojmy vysvětlené v kapitole 2. 3. 1. V ER diagramu se zakreslují objekty podle obrázku 10.



Obrázek 10 Popis ER diagramu (export z MySQL Workbench)

#### Relace

Relace je vazba mezi tabulkami. Pomocí relace znázorňujeme kardinalitu a povinnost členství. Graficky se vazby řeší podle obrázku 1.



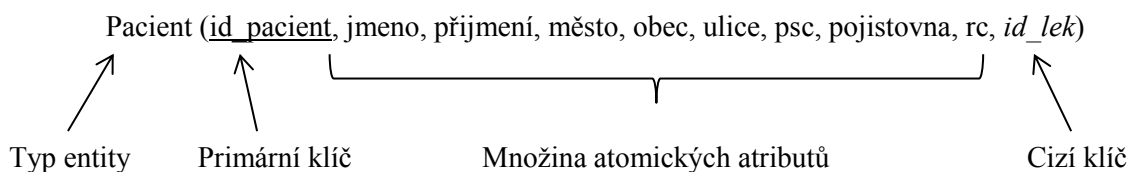
Obrázek 11 Obrázek 10: Typy relací. Plnou čarou se značí povinnost členství, čárkovanou nepovinnost.

### 4.3 Logický návrh databáze

Logický návrh je v podstatě uvědomění si spojitostí nashromážděných dat. Vycházíme vždy z konceptuálního návrhu. Data se vkládají do tabulek s definovaným omezením. V tomto kroku se přihlíží na transakce a různé pohledy. Vytváří se logická struktura databáze.

#### 4.3.1 Lineární zápis

Lineární zápis slouží k zápisu množiny atributů jednoho typu entity. V množině atributů se vyznačuje primární a cizí klíč. Zápis vypadá takto:



#### 4.3.2 Datový slovník

Úplné tabulky atributů. Datové vyjádření atributů, které podrobně popisují atributy. Mají jasně vyčleněný zápis, kterým popisují realitu.

Tab. 4: Ukázka datového slovníku

Tabulka	Atribut	Datový typ	délka	klíč	NULL	index	IO	Význam, poznám
Pacient	id_pacient	INT	10	A	N			Auto inkrement
	jmeno	WARCHAR	45	N	A			Doplní sestra
	prijmeni	WARCHAR	50	N	A			Doplní sestra
	Id_lek	INT	10	N	A			Cizí klíč
	...							

### 4.4 Funkční analýza

Funkční analýza nebo model je ujasnění logické struktury a implementování struktury k uživateli. Je zapotřebí vytvořený systém přizpůsobit uživateli. Vytvořit rozhraní a uvědomit si aplikační procesy. K tomu slouží mnoho grafických diagramů. V bakalářské práci byly použity diagramy datových toků (Data Floud Diagram), UCD (Use Case Diagramy), a také diagramy aktivit.

#### 4.4.1 Diagram aktivit

Diagram aktivit je jedním z UML diagramů k popisu chování systému. Pomocí akčních symbolů a zachování správných toků se popisují přehledně procesy.

##### Akční symboly

Symbole popisující zlomovou situaci (popisují děj). Můžeme rozdělit na:

- akce – atomické dále nedělitelné kroky
- aktivity – obecný druh popisu umožňující uchovávat vlastní diagram aktivit

#### Tok aktivit

Tok znázorňuje sekvenci mezi ději. Důležité je dodržet tok do aktivity a tok z aktivity. Nikoliv nejde, že všechny toky vycházejí z aktivity, jelikož tok vždy začíná v začátku a končí na konci (výjimečně v předčasném konci).

#### Rozhodnutí

Proces, kdy dochází k rozdělení toku pod událostí rozhodnutí ANO, NE. Nad symbolem rozhodnutí je vždy zapotřebí uvést otázku proč? A odpovědi znamenat ANO, NE.

#### Spojení

Pro spojení toku se vždy doporučuje speciální symboly, ať už je to spojovací člen, který spojuje mezi sebou nezávislé toky, nebo jsou to přímo aktivity. Aktivity jsou specifické spojení, jejímž prostřednictvím je nahrazován logický člen AND.

### **4.4.2 Diagram Datových toků**

DFD (Data Flow Diagram) je nástroj k podpoře spíše konceptuálního modelování, ale poukazuje na funkční zpracování. DFD je vhodný pro ukázkou datových toků systému. Má základní terminologii a hierarchii. Může se začít obecně a postupně jít do hloubky systému.

#### Proces

**Proces** (funkce, transformace – aj. Proces) představuje část systému, která mění vstupy na výstupy. Symbolem procesu je kruh, ovál nebo obdélník (dle druhu notace). Proces je nazýván jedním slovem, krátkou větou nebo frází, která má jasně vyjadřovat jeho podstatu. Do procesu se zavádí číslování hierarchie. [6]

#### Tok

**Tok** znázorňuje přesun informace z počátečního bodu do bodu cíleného (do procesu).

#### Terminátor

**Zdroj/určení (terminátor)** představuje externí zdroj nebo místo určení dat. Jsou tedy mimo systém a určují původ a cíl dat. Data musí pocházet z jednoho nebo více zdrojů vně a musí směřovat do jednoho nebo více určení. Zdrojem nebo určením může být cizí organizace nebo její část, jiný informační systém nebo osoba (např. zákazník), která je v interakci se systémem. Zdroj/určení je označován obdélníkem s vepsaným názvem.

#### Uložiště dat



*Úložiště dat představuje místo, kde jsou uložena data. Může být implementován různě (jako pole, soubor – obyčejný nebo databázový, ale i jako cokoliv jiného, například šanon, kniha apod.). Používá se všude tam, kde mezi procesy existuje časově zpožděné předávání dat (asynchronní). Asynchronnost procesů musí vyplývat vždy z jejich podstaty, nikoliv z formy implementace (to je záležitostí technologického nebo implementačního modelu). [10]*

#### **4.4.3 Diagram užití**

Diagram užití pomáhá dívat se na systém z pohledů uživatele a systému. Nástroj UML pro objasnění problematiky s výbornou vypovídající schopností. V diagramu užití se používají pro znázornění tři základní značení.

##### Aktér

Symbol pro zobrazení extérních objektu vstupujících do vztahu procesů. Může reprezentovat jak osobu, tak i charakteristický objekt či veličinu.

##### Příklady užití toku:

- Precedes → Předchází
- Invokes → Vyvolá
- Realise → Uvědomte si
- Extend → Rozšíření
- Include → Zahrnout
- Generalize → Obecně (vazba mezi aktéry)
- Associate → Spojovat
- Use → Používá

##### Případ užití

Případ užití reprezentuje část funkcionality vyšší úrovně poskytované systémem, která tvoří logický celek. Soubor případů užití tak dovoluje zákazníkovi pohled na celý systém. Lze jej definovat jako soubor scénářů, které spojuje společný cíl. Počet případů užití by měl být vyvážený a měly by kompletně popsat systém. (<http://orca.xf.cz/ooms/007/007m.htm>)

## 5 Praktická část bakalářské práce

Tato část bakalářské práce se zabývá návrhem databáze, pro urgentní příjem fakultní nemocnice Ostrava. Databáze byla vyvíjena v databázovém systému My SQL worbench verze 5.5. Tento systém je v angličtině ale nabízí zajímavější uživatelské prostředí než např. PhpMyAdmin. Data budu ukládat na školní server, přístupný na [rfidexpert.vsb.cz](http://rfidexpert.vsb.cz). Server je v provozu 24 hodin denně.

Bylo nezbytné zajistit povolení nacházet se na urgentním příjmu jelikož, nezdravotní osoby na místech jako je urgentní příjem nemají co dělat. Povolení se protáhlo a tak na důkladnou analýzu urgentního příjmu zbyl malý časový interval.

Bylo zapotřebí zjistit způsoby vedení evidence a jednoznačně vy cílit informace o pacientu, přístroji, lécích atd. Byl vysledován pohyb pacienta a manipulace s pacientem. Byly zjištěny postupy a na základech těchto postupů byla navrhnutá databáze.

V této části jsou použity objemné diagramy, které jsou nezbytné pro vypovídající hodnotu práce. Datový slovník vypovídající o evidenci k ZIP.

## 6 Analýza urgentního příjmu fakultní nemocnice Ostrava

Analyzovat prostředí urgentního příjmu bylo zásadním úkolem pro vytvoření smysluplné databáze. Urgentní příjmy nemocnic v České republice nemají standardizované postupy v ošetření pacienta. Vlastně každý urgentní příjem má jiné prostory a jiné postupy.

V analýze bakalářské práce byly popsány jednotlivé postupy uplatňující se na urgentním příjmu FNO. Velmi důkladně bylo přihlíženo na pacienta a jeho pobyt na urgentním příjmu. Od ohlášení příjezdu až do propuštění pacienta. Jednotlivé výkony byly popsány obecně, jelikož analyzovat každou situaci by bylo časově náročné ne-li nemožné.

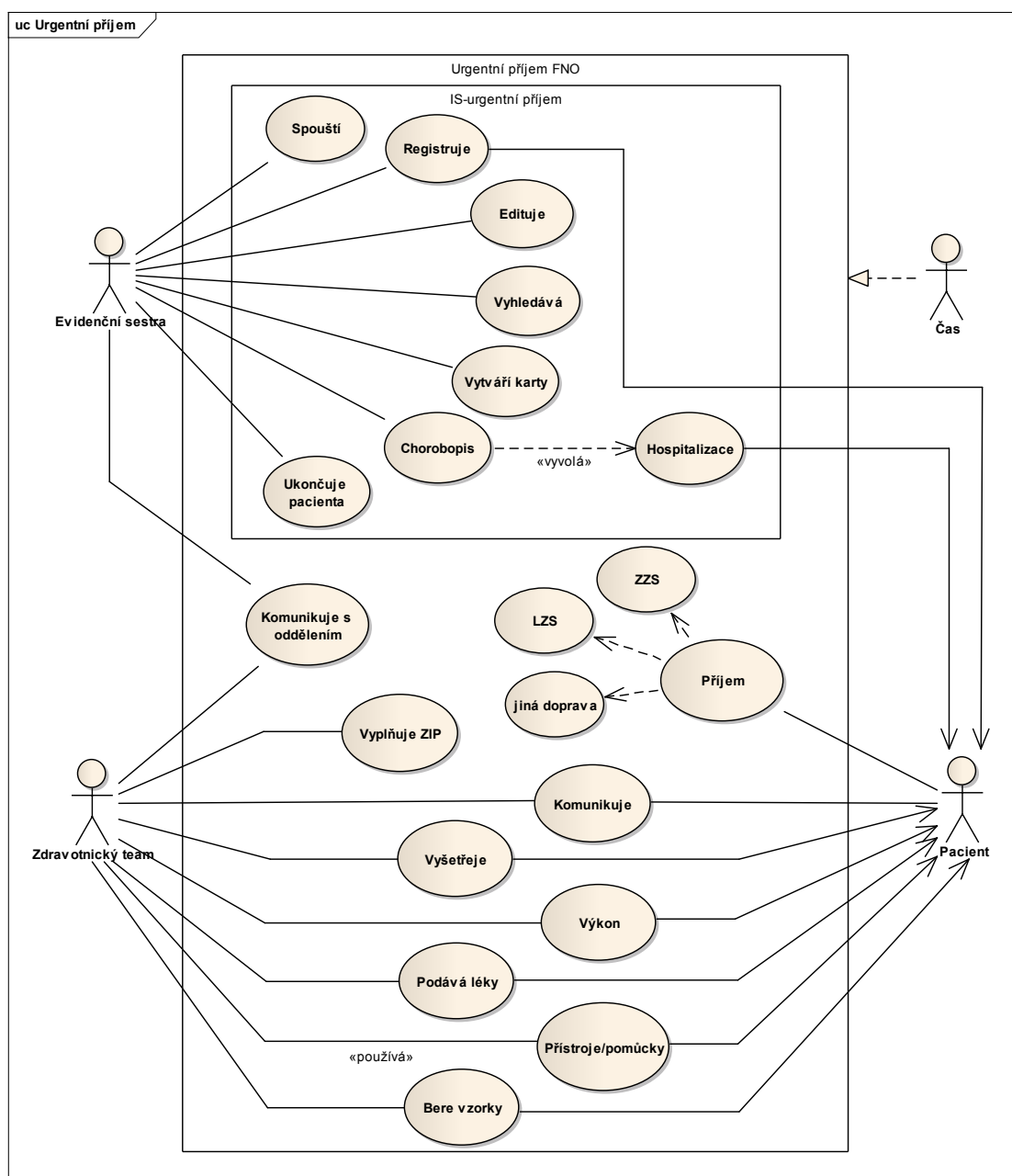


Diagram 1: Diagram užití použitý na urgentní příjem FNO znázorňuje přehled práce.

## 6.1 Definování základních pojmů

**Pacient** – klient urgentního příjmu s potřebou ošetření

**Evidenční sestra** – Osoba pověřená práci s IS a s evidencí pacienta

**Zdravotnický team** – zdravotnický personál, který vykonává práci na pacientovi

**Čas** – Velmi důležitá veličina

## 6.2 Personál

Na urgentním příjmu fakultní nemocnice momentálně pracuje zdravotnický team složený z celkem z 13 zaměstnanců. Na denní směně je to 5 zdravotních sester, včetně staniční sestry. Dva záchranáři, jeden zdravotnický asistent a jeden sanitář. Samozřejmě je lékař a po případě i lékaři z jiných oddělení. Tito všichni se podílejí na pacientovi.

## 6.3 Informační systémy na urgentním příjmu FNO

Na urgentním příjmu FNO, se používají pro registraci a vedení záznamu o pacientech dva informační systémy. CLINICOM, který využívají především sestry a pro doktory CARE CENTUM. Tyto informační systémy jsou vedeny přes šifrovací komunikační protokol s nemocničním informačním systémem. Tedy dokáží grupovat informace o pacientovi. Pokud se pacient ocitne ve FNO a bude registrován, pak se jeho registrace uloží a je možné, si jej kdykoliv vyhledat a doplnit.

Pomocí, dvou zmíněných systému, evidenční sestra provádí jednotlivé úkony. Vyhledává, registruje, eviduje, maže a vytváří karty, nebo chorobopisy. Všechny tyto úkony jsou na jednoduchém uživatelském prostředí, který od začátku, provede uživatele, až na vytištění dokumentu. Na tyto systémy se klade velký nárok na malé zatížení uživatele (kdyby se to přehnalo, uživatel si vystačí pouze s myší).

## 6.4 Postup pacienta urgentním příjmem

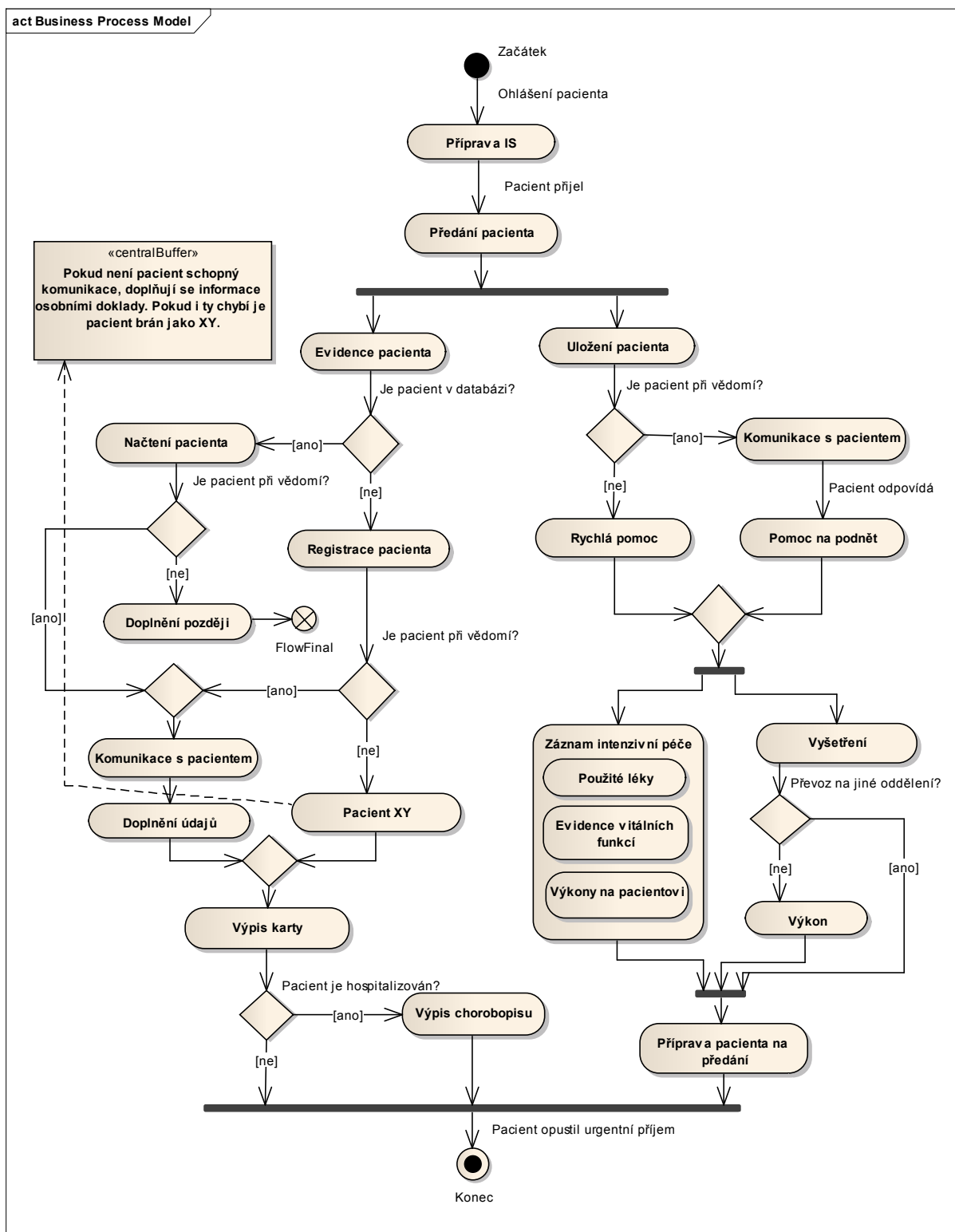


Diagram 2: Obecný přehled práce na urgentním příjmu.

#### **6.4.1 Přijetí pacienta**

Přijetí pacienta v diagramu číslo 2 reprezentují dva začáteční bloky. V diagramu číslo 1 pak případ užití příjem.

Na urgentní příjem fakultní nemocnice Ostrava se pacient může dostat několika způsoby. Nejčastěji to bývá zdravotní záchranou službou (ZZS). Ale může to být leteckou záchrannou službou (LZS), nebo vlastní dopravou. Velmi zřídka je pak pacient přeposlán z interního příjmu na příjem urgentní ale i to se stává.

Způsoby pacientova přijetí:

- Záchraná zdravotnická služba (ZZS)
- Letecká zdravotnická služba (LZS)
- Vlastní doprava
- Z jiného oddělení

Pokud pacient dorazí ZZS nebo LZS jsou to ohlášené případy. Evidenční sestra dostane zprávu o příjezdějším pacientovi. Dozvídají se stav pacienta a jeho počáteční diagnózu. Zahájí přípravu na pacienta. Nachystá se dokumentace a potřebné zdravotnické pomůcky na ošetření. Uvedeno na příkladu. Na centrále urgentního příjmu dostávají záznam o převozu pacienta. Zpráva se předá týmu, který začne s přípravou manipulačního lůžka. Evidenční sestra se stará o interní dokumentaci tzn., napíše číslo pacienta, nachystá záznam intenzivní péče (ZIP), spouští IS a informuje o poloze pacienta, jelikož otevírá garáž (místo příjezdu ZZS), nebo dveře k heliportu (LZS). Tyto místa jsou monitorovány videotechnikou. Velkou výhodou těchto převozů je včasné varování a komunikace se zdravotní službou.

V ostatních případech je to situace nepředpokládaná a málo pravděpodobná.

#### **6.4.2 Uložení pacienta**

Po převzetí od záchranné služby se pacient uloží na manipulační lůžko a je odbaven. Paralelně se eviduje nový pacient v IS (clinicom). Velkou roli v těchto procesech hraje vědomí pacienta. Vědomí je základní rozdílový prvek s prací na pacientovi. Pokud pacient komunikuje a spolupracuje, je snazší jeho ošetření. Riziko vulgárních pacientů a nepřizpůsobivých lidí, kteří komplikují práci zdravotnického týmu, se v praxi bohužel vyskytuje často. Po převlečení a očištění se pacient převezí na jedno ze dvou interních lůžek. V případě dítěte se převáží do speciálního boxu pro dětské pacienty, kde jsou také dvě lůžka. V případě potřeby jsou k dispozici další dvě lůžka ve speciálním uzavřeném boxu.

### 6.4.3 Evidence pacienta

Evidence pacientů se provádí za účelem vedení záznamu a výkazu pro zdravotní pojištění. Evidenci pacienta má na starost evidenční sestra, která pacienta zadává do IS. Dále dělá interní vedení záznamu. Což je ve skutečnosti přiřazení pacientovi číselný kód, který se vždy inkrementuje o jednu nahoru. Pro evidenci pacientů, se používá CLINICOM. Tento systém vyhledá v nemocničním systému pacientovu kartu. Pokud najde, doplní sám potřebné informace. Evidenční sestra doplní aktuální informace. Pokud nenajde je zapotřebí pacienta registrovat. Tato práce je znázorněna na diagramu 2 od větve evidence pacienta.

Základní evidované informace k pacientovi:

- Identifikační číslo Pacienta
- Rodné číslo
- Jméno a příjmení
- Adresa
- Pojišťovna
- Čas příjmu, čas předání pacienta

#### Pacient XY

Vedení pacientova záznamu XY se provádí tehdy, kdy pacient není schopný komunikace a nemá žádné osobní doklady. Pak je brán jako pacient XY a do záznamu se píše datum, čas příjmu, pohlaví a místo odkud byl přivezen po případě charakteristika osoby (např. cyklista).

### 6.4.4 Záznam intenzivní péče

Příklad záznamu intenzivní péče (ZIP) je možno vidět na diagramu 2 a 3. Je to interní záležitost FNO. Způsob papírové evidence práce na pacientovi. Formulář velikosti A3 přehledně strukturalizován. Do formuláře se zapisují smluvenými symboly v časové linii např. léky, vitální funkce tzn. krevní tlak, puls ale také výkony, laboratorní vzorky, krevní banka, konzilia, výška, váha, bolest a mnoho dalších parametrů, které jsou později uvedeny. Tento záznam se přikládá ke kartě pacienta a putuje s ním na další oddělení.

### 6.4.5 Smrt

Smrt pacienta je speciální situace, která může nastat v každé fázi procesu. Je to neočekávaná událost, kterou se snažíme předejít. Když nastane smrt, je evidován čas smrti. Na pacientovi přestanou zdravotnické výkony (zdravotnický team má povinnost dodržovat státem nařízené postupy oživování) a pacient se převezí do místnosti MPZ. Dále se musí neprodleně kontaktovat rodina a vypsát karta.

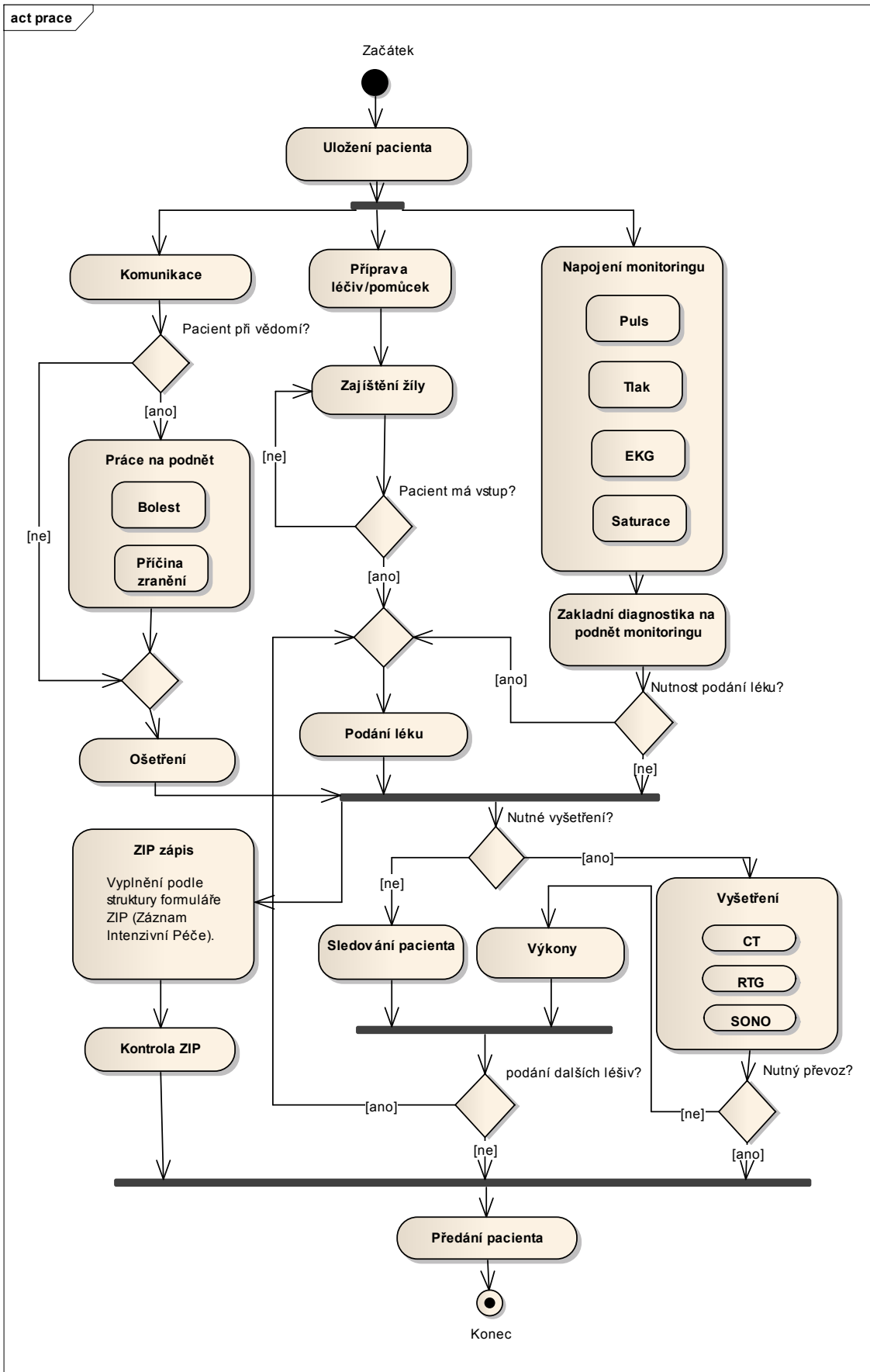


Diagram 3: Proces práce na pacientovi.



## 6.5 Analýza identifikátoru

Analýzou identifikátoru se rozumí rozřídít jednotlivé entity na ty, které se budou moct načítat speciálním identifikačním kódem a na ty, které identifikátorem nejdou označit. Ty, které označit nejdou, jsou většinou veličiny např. tep, krevní tlak, teplota, saturace a další přístroji změřená data. Tyto data sice označit nejdou, ale dali by se ukládat na datový disk a využít pomocí internetu, jako automatický záznam k pacientu. Tím by se eliminovali chyby lidského faktoru, ale naproti tomu by to muselo mít vysoké nároky na bezpečnost plus systémové opatření. Ostatní entity, které nejdou označit a také nejsou diskrétně měřeny, nezbyvá, než zapsat do systému ručně. Ty entity, které nejsou značeny a značit se můžou (např. vyšetření) se označí speciálním identifikačním číslem. Tedy každému druh vyšetření se přidělí jeho identifikační kód.

### 6.5.1 Typy identifikátoru

Ne každý prvek je označený stejným typem identifikátoru. Léky jsou nejčastěji značeny na krabičce svým čárovým EAN kódem. Vyskytují se, ale léky které jsou značeny QR kódem. Nejlepší možný způsob identifikátoru je však RFID. RFID je identifikátor komunikující pomocí frekvencí a svůj kód doslova vyzařuje, tudíž se nejrychleji čte a je vlastně minimálně náročný na obsluhu čtení. Identifikátory se čtou pomocí média (např. mobilní telefon, čtečka kódu...).

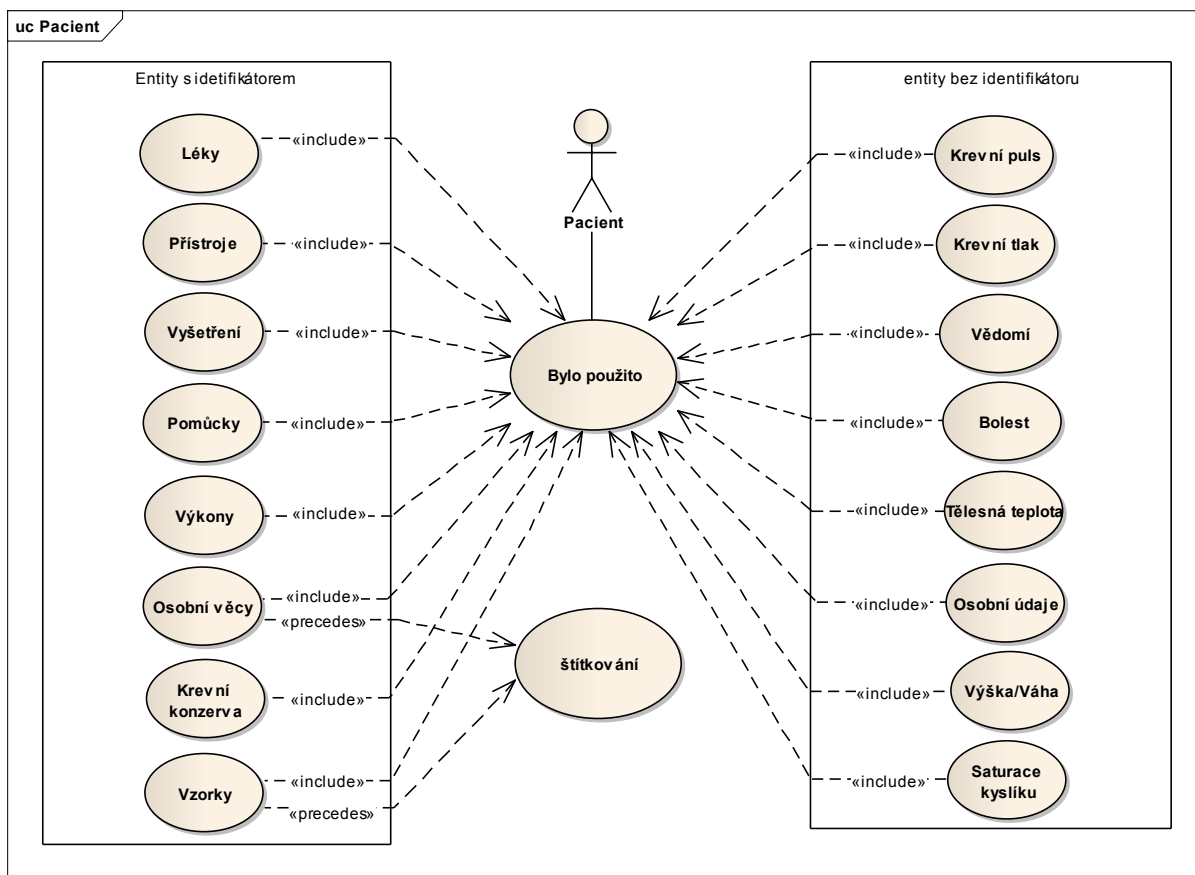


Diagram 4: Roztřizeny informace o označených entitách

## 7 Návrh databáze

Databáze byla navržena tak, aby splňovala koncept internet of things. To znamená, že bylo zapotřebí vytvořit místa pro identifikační značení a hlavně vymezit jednotlivé entity, které se dají označit. První návrh databáze byl takový, že se vytvoří speciální tabulka, do které budou směřovat všechny tabulky, které by mohli být použity. To znamená, speciální tabulka by byla unikátní pro jednoho pacienta a unikátní pro jakýkoliv použitý objekt k pacientovi. Bohužel tento návrh neuspěl v praxi a pro záměrné určení objektu byly vytvořeny vazební tabulky k pacientovi. Tento návrh je poněkud obsáhlejší pro databázi, právě o vazební tabulky, ale výhodnější pro načítání identifikátoru a následného uložení do databáze.

### 7.1 Konceptuální modelování

Konceptuální modelování, bylo založeno na informacích získaných od pracovníků urgentního příjmu. Byl sestavený seznam tabulek a jejich charakteristik. Navržen E-R diagram a specificky popsány atributy.

#### 7.1.1 Lineární zápis

U pacienta evidujeme tyto atributy:

Pacient (id\_rc, jmeno, prijmeni, kod\_pojistovny, mesto, obec, ulice, cp, psc, datum\_narozeni, pohlavi, zamestnani, tel\_blizka\_osoba, nabozenske\_vyznani, cas\_nacteni, kod\_pacienta)

Typy entit označené koncovkou ZIP jsou entity evidované ve formuláři ZIP:

Hlavicka\_ZIP (id, skala\_bolesti, komunikace, teplota\_tela, drogy, alkohol, vyska, vaha, alergie, cas\_ukonceni\_pacienta, *Pacient\_id*)

Leky\_ZIP (id, kod\_leku, mnozstvi, nazev, zpusob\_podani, podnazev, poznamku, typ)

Vykon\_ZIP (id, nazev, kod\_vykonu, poznamka)

Krevni\_banky\_ZIP (id, kod\_konzervy, lhuta, druh, krevni\_skupina, mnozstvi, poznámka)

Vysetreni\_ZIP (id, kod, nazev, poznamka)

Konzilia\_ZIP (id, kod\_konzilia, nazev)

Laborator\_ZIP (id, nazev, kod, podnazev, poznamka)

Typy entit, které se mohou v ZIP vyskytovat, ale ve výsledku přesahují jeho rámec:

Osobni\_vecy (id, cennosti, místo\_ulozeni, datum, obleceni, prirucni\_zavazadlo, kod)

Vzorek (id, kod\_vzorku, *Pacient\_id*)

Zamestnanci (id, kod, jmeno, prijmeni, telefon, pozice)

Revize\_pristroju (id, datum\_revize, *Pacienti\_id*, *Zamestnanci\_id*)

*Pristroje* (id, kod, evidencni\_číslo, název, lokalizace)

Lineární zápis vazebních tabulek:

Pacient\_has\_leky\_ZIP (id, cas\_nacteni, *Pacient\_id*, *Leky\_ZIP\_id*)

Pacient\_has\_Vykon\_ZIP (id, cas\_nacteni, *Pacient\_id*, *Vykon\_ZIP\_id*)

Pacient\_has\_Konzilia\_ZIP (id, cas\_nacteni, *Pacient\_id*, *Konzilia\_ZIP\_id*)

Krevni\_banku\_ZIP\_has\_Pacient (id, cas\_nacteni, *Pacient\_id*, *Krevni\_banku\_ZIP\_id*)

Pacient\_has\_Vysetreni\_ZIP (id, cas\_nacteni, *Pacient\_id*, *Vysetreni\_ZIP\_id*)

Pacient\_has\_Pristroje (id, cas\_nacteni, parametry, *Pacient\_id*, *Pristroje\_id*)

Pacient\_has\_Osobni\_vecy (id, cas\_nacteni, *Pacient\_id*, *Osobni\_vecy\_id*)

Konzilia\_ZIP\_has\_Zamestnanci (id, *Konzilia\_ZIP\_id*, *Zamestnanci\_id*)

*Laborator\_ZIP\_has\_Vzorek* (id, cas\_nacteni, *Laborator\_ZIP\_id*, *Vzorek\_id*)

### 7.1.2 Datový slovník

Důležitá část konceptuálního modelování. Kvůli rozsahu byly uvedeny pouze datové slovníky pro pacienty a pro tabulky ZIP. Vazební tabulky byly vloženy do příloh.

**Tab. 5: Datový slovník charakterizující atributy v tabulce pacient.**

Typ Entity	Atribut	Datový typ	Délka	Klíč	NUL	IO	Poznámka
<b>Pacient</b>	id	INT	100	A	N	Primární klíč	Auto. navýšení
	rc	BIGINT	10	N	A		rodné číslo
	jmeno	VARCHAR	45	N	A		jméno
	prijmeni	VARCHAR	45	N	A		příjmení
	kod_pojistovny	INT	5	N	A		adresa
	město	VARCHAR	100	N	A		adresa
	obec	VARCHAR	100	N	A		adresa
	ulice	VARCHAR	100	N	A		adresa
	cp	INT	11	N	A		adresa
	psc	INT	5	N	A		adresa
	datum_narozeni	DATE	-	N	A		
	pohlavi	ENUM *	-	N	A		*('m', 'z', 'n')
	zamestnani	VARCHAR	100	N	A		
	tel_blizka_osoba	INT	9	N	A		
	nabozenske_vyznani	VARCHAR	100	N	A		
	cas_nacteni	DATETIME	-	N	N		čas příjmu pacienta
	kod_pacienta	VARCHAR	100	N	A		QR, EAN, RFID

\*A= ANO, N= NE.

**Tab. 6: Datový slovník entit vyhrazených pouze jako ZIP.**

Typ Entity	Atribut	Datový typ	Délka	Klíč	NULL	IO	Poznámka
<b>Hlavicka_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	skala_bolesti	INT	10	N	A	vědomí	od 0 - 10
	komunikace	VARCHAR	45	N	A	vědomí	
	teplota_tela	VARCHAR	45	N	A		
	drogy	VARCHAR	50	N	A		test na drogy
	alkohol	VARCHAR	50	N	A		test na alkohol
	vyska	INT	5	N	A	vědomí	
	vaha	INT	5	N	A	vědomí	
	alergie	VARCHAR	45	N	A	vědomí	
	cas_ukonceni	DATETIME		N	A		
	Pacient_id	INT	100	A	N	Cizí klíč	
<b>Leky_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	kod_leku	VARCHAR	100	N	N		QR, EAN,
	mnozstvi	INT	10	N	A		Vypotřebované
	nazev	VARCHAR	45	N	N		
	zpusob_podani	VARCHAR	100	N	N		
	podnazev	VARCHAR	45	N	A		
	poznámka	VARCHAR	200	N	A		
	typ	VARCHAR	45	N	A		
<b>Vykon_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	nazev	VARCHAR	100	N	N		
	kod_vykon	VARCHAR	50	N	N		QR, EAN, RFID
	poznámka	VARCHAR	200	N	A		
<b>Krevni_bankazIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	kod_konzervy	VARCHAR	45	N	N		QR, EAN, RFID
	lhuta	DATE		N	N		datum spotřeby
	druh	VARCHAR	45	N	N		
	krevni_skupina	VARCHAR	10	N	N		
	mnozstvi	INT	5	N	N		
<b>Vysetreni_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	kod	VARCHAR	50	N	N		QR, EAN, RFID
	nazev	VARCHAR	100	N	N		
	poznámka	VARCHAR	200	N	A		
<b>Konzilia_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	kod	VARCHAR	100	N	N		QR, EAN, RFID
	nazev	VARCHAR	100	N	N		
<b>Laborator_ZIP</b>	id	INT	100	A	N	Primární klíč	aut. Navýšení
	kod	VARCHAR	100	N	N		QR, EAN, RFID
	nazev	VARCHAR	45	N	N		
	podnazev	VARCHAR	45	N	A		
	poznámka	VARCHAR	200	N	A		

### 7.1.3 E-R diagram



Diagram 5: ER datový model

### 7.1.4 Popis ER diagramu

Objektově relační diagram znázorňuje na diagramu 5 vztahy mezi tabulkami. Dominantní tabulkou v tomto diagramu je Pacient. Tato tabulka eviduje pacienty. Prázdné symboly u atributů vyznačují hodnotu NULL. Plné modré symboly vyznačují NOT NULL. To znamená, že v jednom záznamu tzn. jeden pacient, musí mít vyplněny atributy čas příjmu a kód pacienta. Bez těchto atributů a samozřejmě klíče id (ikonka klíče znamená primární klíč tabulky<sup>3</sup>), by nebylo možné záznam pacienta evidovat. Jelikož přečtená informace z mobilního zařízení je schopná přečíst jen čas a kód (není umožněn zápis jména atd.), je nutné ostatní atributy uvést jako NULL hodnoty.

Tomuto řešení předcházela diagram v příloze D, Původní myšlenka modelu byla vytvořit tabulku, která by byla jedinečná právě pro jednoho pacienta. V té tabulce by se zaznamenávali úkony, spojeny právě s tím jedním pacientem. Podle mého názoru zcela funkční řešení eliminující vazební tabulky. Bohužel tento model nevyhovoval a byl nahrazen tvrdými vazebnými tabulkami. Vazební tabulky sice, rozšířili objem databáze, ale přímo určili jasné použití objektu.

### 7.1.5 Diagram datových toků

Tento diagram byl použit k ujasnění toků informací na pracovišti urgentního příjmu. Diagram nejprve ukazuje svrchní vrstvu systému a potom se snaží jít pod ní, tedy jeho hierarchie má dva stupně.

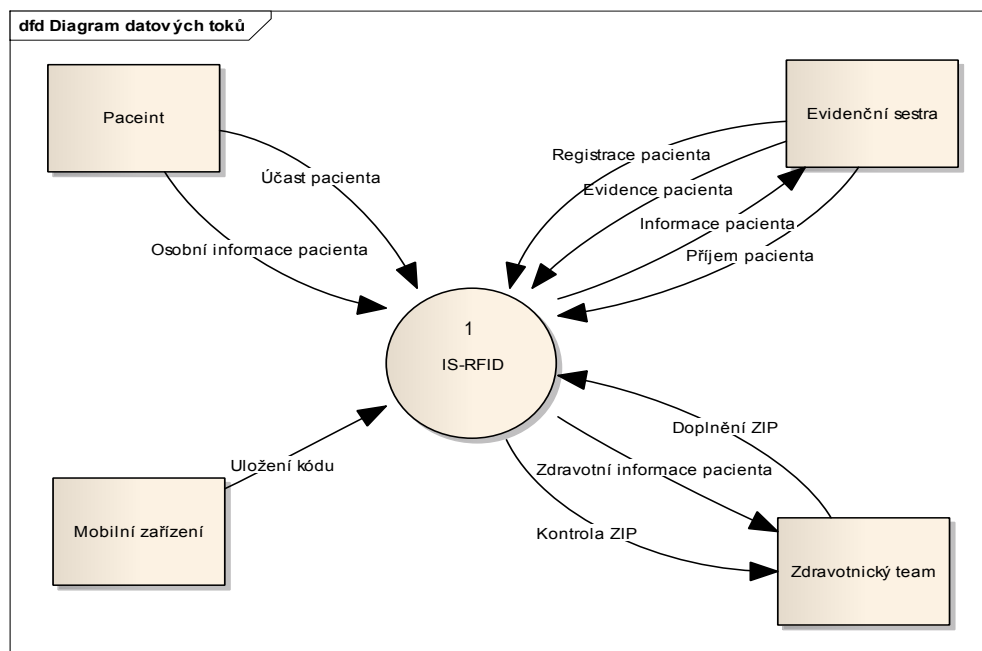


Diagram 6: Diagram datových toků popisující vrchní hierarchii

<sup>3</sup>Nevýhodou značení Workbench, jsou cizí klíče tabulek s povinnou vazbou, které nejsou rozlišeny od primárního klíče.

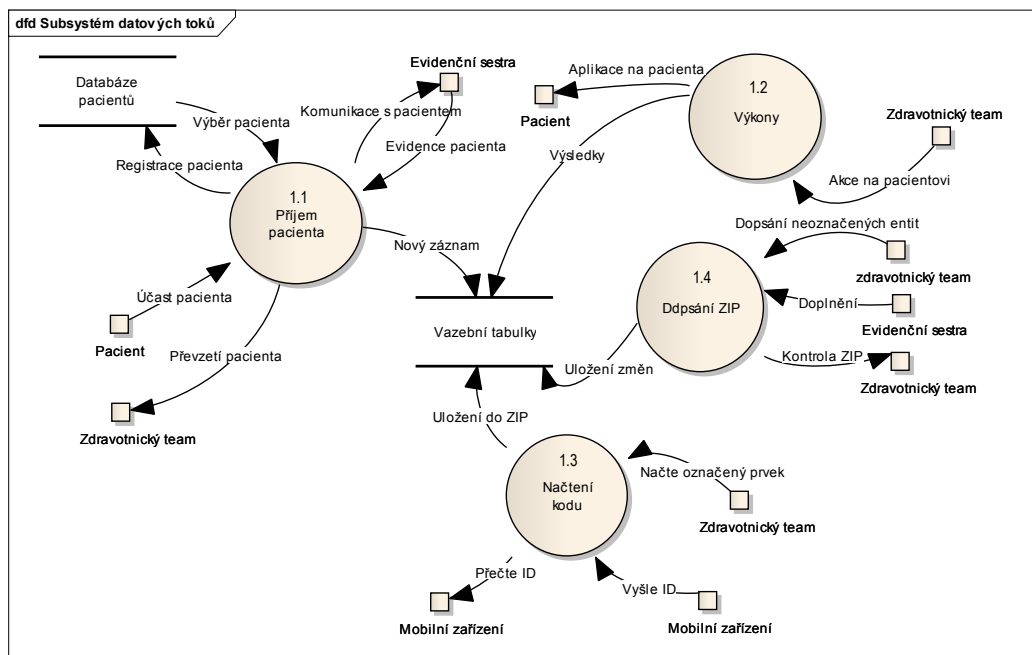


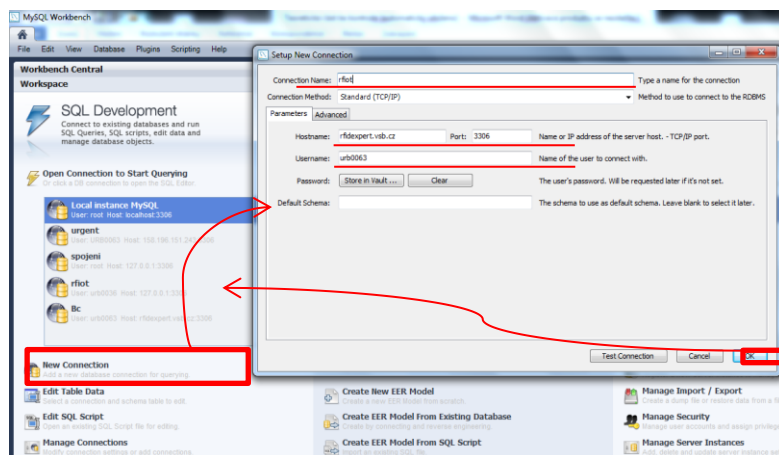
Diagram 7: DFD popisující práci a užití informačního systému v praxi.

## 7.2 Fyzický návrh

Ve fyzickém návrhu byl popsán postup vzniku databáze v systémové aplikaci Workbench. Workbench je anglicky psané uživatelské prostředí pro správu databáze.

### 7.2.1 Připojení databáze

Pro připojení na databázový server byl použit právě MySQL Workbench. Titulní strana tohoto softwaru nabízí správu serveru, modelování ER diagramu a vytvoření lokálního serveru. V práci, jak už bylo zmíněno, byl použit školní server, tudíž pro práci stačilo pouze zadat přihlašovací údaje. Přihlášení databáze je zobrazeno na obrázku 13. Pro přihlášení stačí novou ikonu přihlášení rozkliknout.



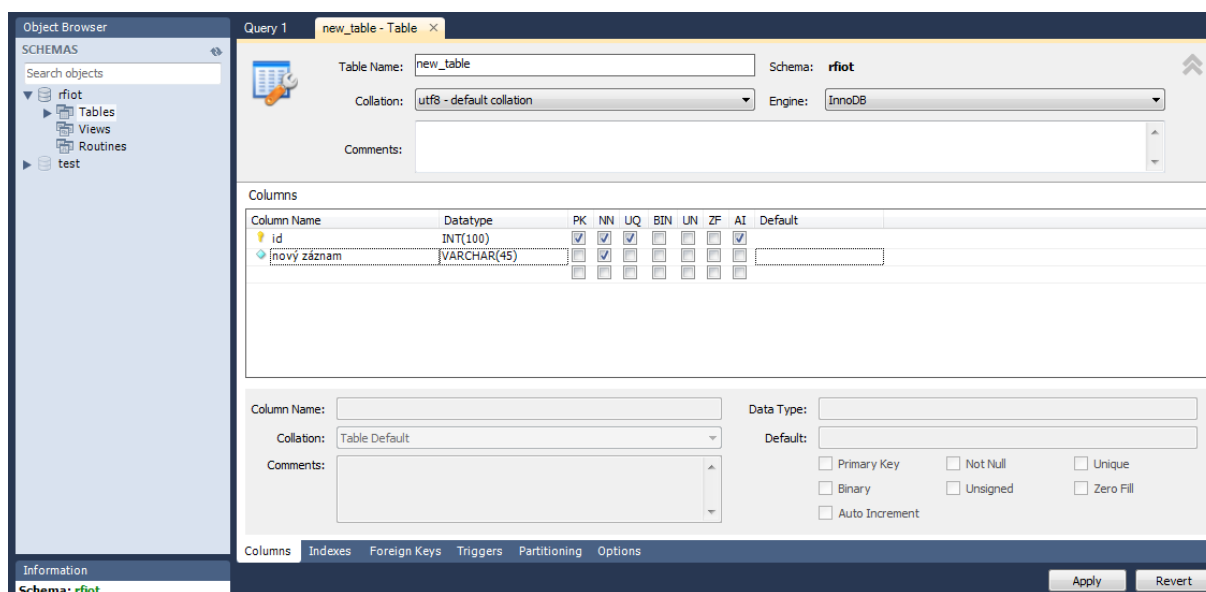
Obrázek 12: Postup nového připojení k existující databázi

### 7.2.2 Vytvoření tabulky

Po připojení k databázi a rozkliknutí připojení se uživatel dostane do uživatelského rozhraní MySQL. V *Object Browser* rozklikneme pojmenované připojení. Vyroloví se tabulky, pohledy a postupy. Pro nabídku vytvoření tabulky, se musí myší kliknout pravým tlačítkem a zvolit *Create Table*. Tímto se dostaneme do průvodce tvorby tabulky, kde postupujeme takto:

- *Table Name*
- *Collation* → *utf8 – default collation*
- *Engine* → *InnoDB*
- *Columns edit*

Pokud se plní tabulka, tak právě v *Columns*. Vlastně pojmenování tabulky definujeme typ entity a vyplnění jednoho záznamu v *Columns* definujeme jeden atribut. Definujeme jeho název, typ, délku, a další omezení zobrazené na obrázku 13. Ve spodní liště je k dispozici další nastavení tabulek. Např. *Indexes* - vyhledávání, *Foreign Keys* - nastavení klíčů, *Triggers* – nastavení pohledů, *Partitioning*, *Options* - možnost uzamykání tabulek.



Obrázek 13: Schéma pro vytvoření tabulky

Vytvoření tabulky se dá i podle zápisu SQL jazyka (jednoduchá ukázka):

- `CREATE TABLE `rfiot`.`Ukazka` (`id` INT(10) NOT NULL AUTO_INCREMENT, `cena` FLOAT(100,2) NOT NULL, `nazev` VARCHAR(45) NULL, PRIMARY KEY (`id`), UNIQUE INDEX `id_UNIQUE` (`id` ASC))`
- `ENGINE = InnoDB`
- `DEFAULT CHARACTER SET = utf8;`



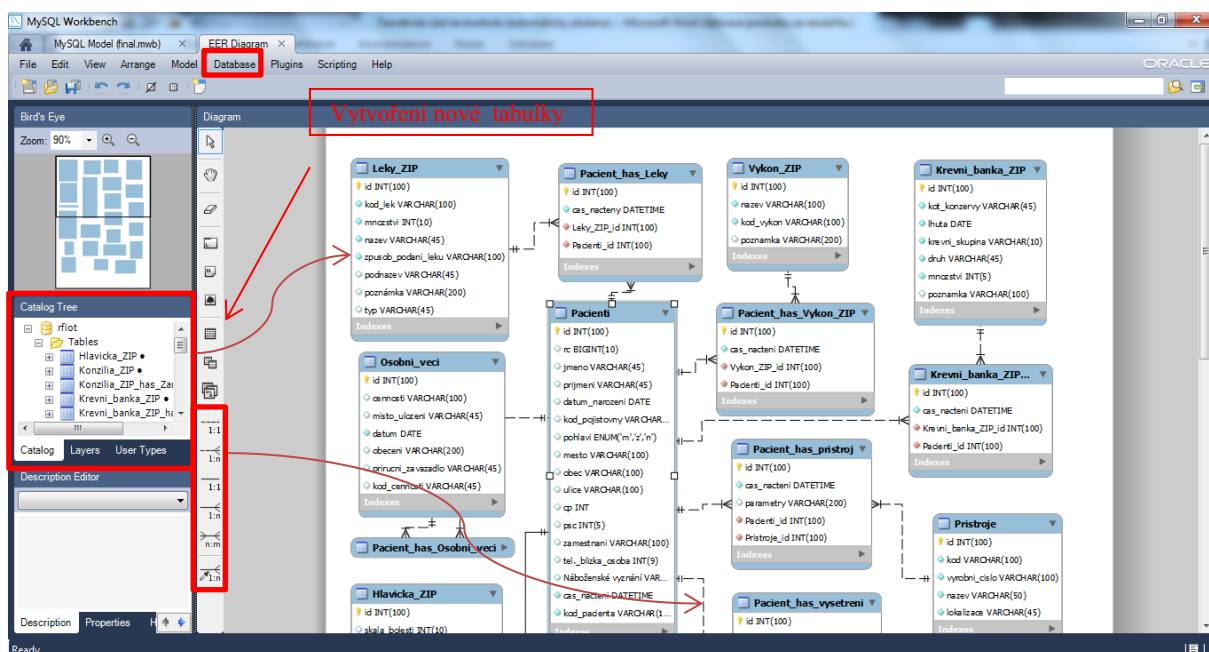
### 7.2.3 Modelování v MySQL Workbench

Pokud byly vytvořeny všechny tabulky a správně vyplněny, vytváří se er model dat. Pro takové modelování je nezbytné se vrátit na titulní stranu (kliknutí na obrazec domu v levém horním rohu). Navolit, vytvoření ER modelu z existující databáze (*Create EER Model From Existing Database*). Následně se uživatel pro kliká průvodcem synchronizování databáze s modulem (nahrají se vytvořené tabulky do modelu). Pomocí *Add Diagram* se vytvoří pracovní plocha. V objektu *Catalog Tree* najdeme importované tabulky z databáze. Přetáhnutím na pracovní plochu se nám graficky ukáží v modelu. Přetáhnutím všech tabulek a uspořádáním jej můžeme logicky propojit vazbami.

V této fázi se dají tabulky různě modelovat i vytvářet. Nutností je však synchronizace modelu s databází. Což často vede ke sporným situacím a nepochopitelným chybám. Je vždy lepší si pečlivě databázi naplánovat a v posledním kroku implementovat do systému. Velkou výhodou tohoto modelování je schopnost relací automaticky určit cizí klíče a tím vytvořit správné vazby. Ovšem pokud se vazba vymaže, mnohdy cizí klíč zůstává v tabulce a je poté zdrojem chyby.

Synchronizaci databáze se provádí následovně:

- Kliknutí na záložku *Database*
- *Synchronize Model (Ctrl+Shift+Z)*
- Pro klikání ke zdárnému konci...



Obrázek 14: Modelování databáze ve Workbench

### 7.3 Test databáze

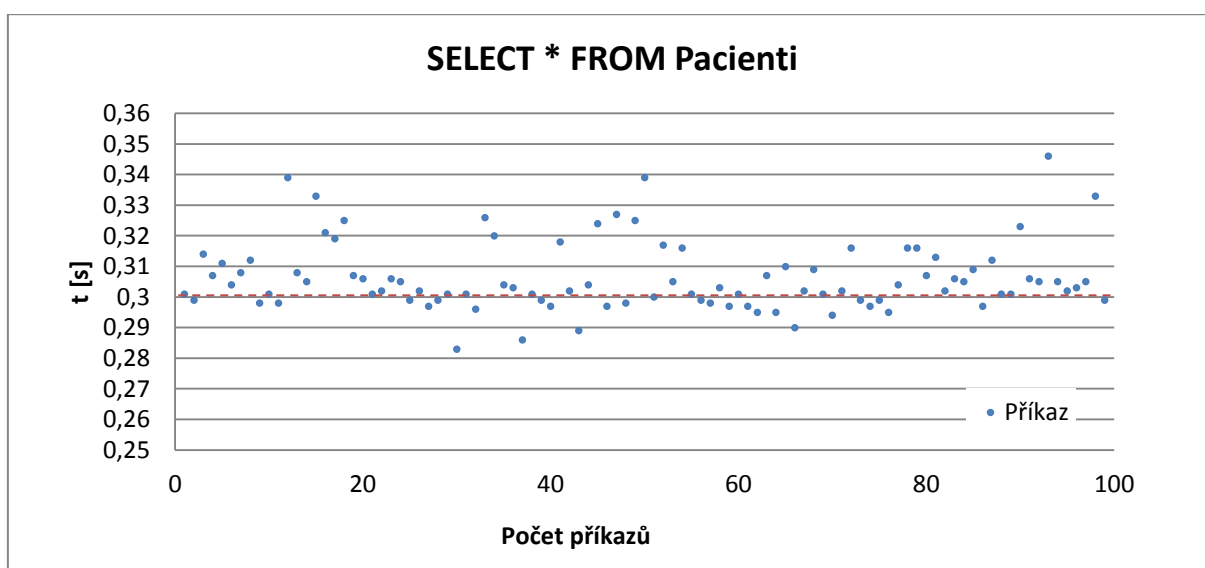
Test databáze se prováděl na výkonném počítači s operačním systémem Windows 8 a parametry:

- Intel i7 2,6 GHz
- HDD 1TB
- RAM 16GB

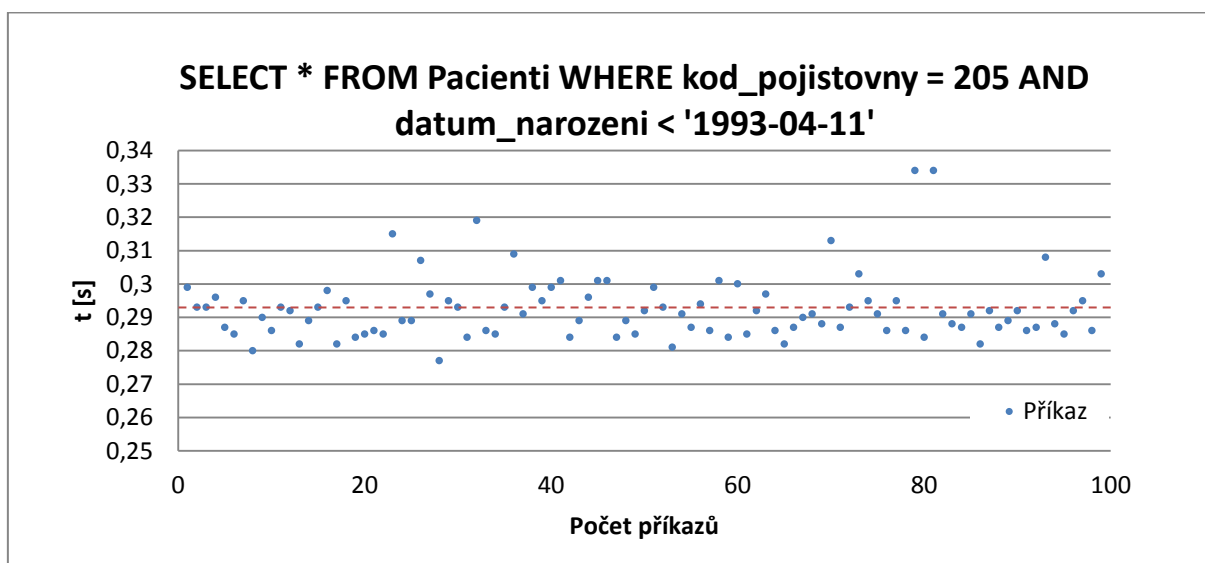
Byly provedeny dva dotazy po 99 opakování:

- `SELECT * FROM Pacient`
- `SELECT * FROM Pacienti WHERE kod_pojistovny = 205 AND datum_narozeni < '1993-04-11'`

Počet vzorků nad, kterými se test prováděl, byl 21 000 záznamu.



Graf 1: Graf znázorňující 99 stejných příkazů jdoucích za sebou a zaznamenávající rychlost odezvy databáze.



Graf 2: Graf znázorňující 99 stejných příkazů jdoucích za sebou a zaznamenávající rychlost odezvy databáze.

## 8 Zavedení databáze

V praxi by zavedena databáze potřebovala zásah do stávajícího prostředí urgentního příjmu FNO. Hlavní požadavek by bylo zajištění bezdrátové internetové sítě po celé ploše pohybu pacienta. Bylo provedeno měření v rámci projektu RRT traumatologické centrum, které vykazovalo nepokryté území. Dále by bylo zapotřebí označení entit identifikátory a to nejlépe RFID. V neposlední řadě optimalizovat čtení a provést průběžné testování a vylepšení databáze. V prostoru by se měli nacházet účelně umístěné čtečky kódu, které by doplňovali mobilní zařízení.

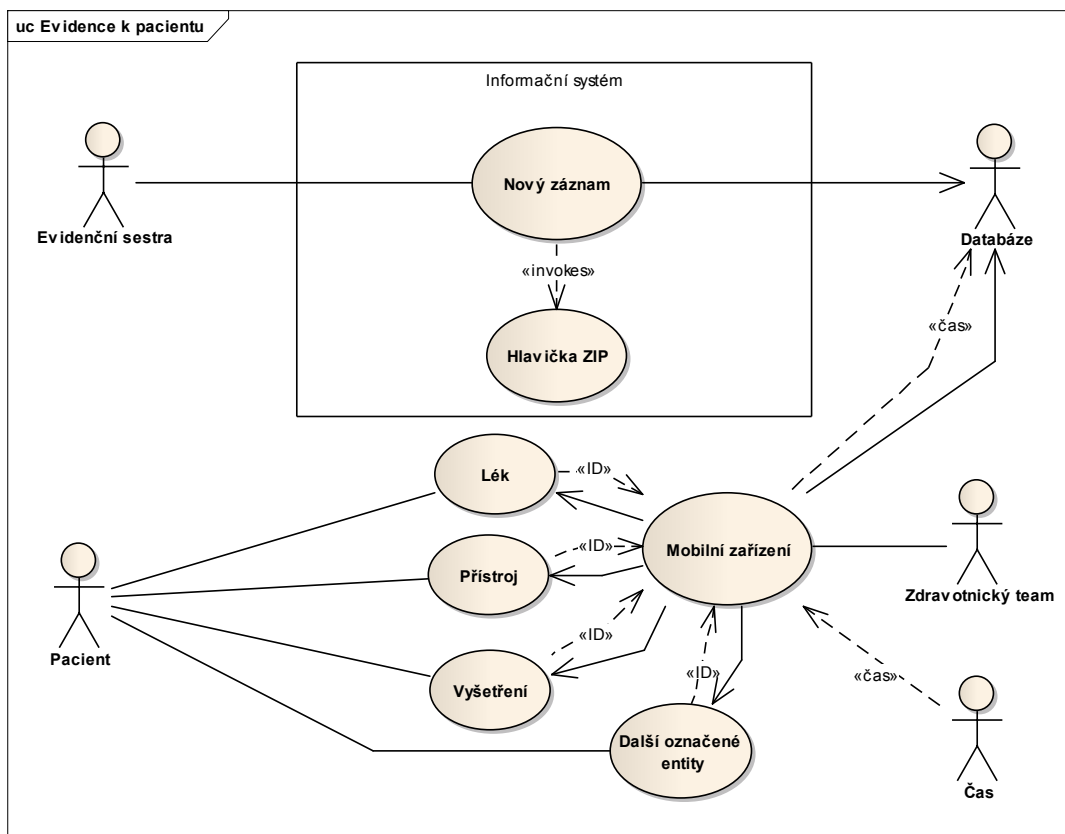


Diagram 8: USE CASE diagram ukazuje na funkčnost systému

Z diagramu je patrná funkčnost systému. Evidenční sestra vytváří nový záznam. Zdravotnický tým k tomuto záznamu shromažďuje data. Přičemž se hlídá čas. Další diagram příloha E.

### 8.1 Funkční princip

Mobilním zařízením se přečte kód. Ten se přes internetovou síť uloží do vazební tabulky databáze spolu s identifikačním kódem pacienta.

Pacienti			
id	jméno	příjmení	kod
1	Jan	Novák	122332
2	Dominika	Pinkasová	456123

Přístroj, který se použil na pacienta			
id	čas	kod_přístroje	kod Pacienta
1	12:54:08	4566de	456123
2	1:25:50	dier88	122332

Přístroje			
id	nazev	kod	lokalizace
1	pumpa	4566de	manipulace
2	defibrilátor	dier88	trauma lůžko

Obrázek 15: Ukázka toku dat

### 8.1.1 Mobilní zařízení

Mobilní zařízení může být každý *smartphone* s operačním systémem Android 2.4 a vyšší. Pro aplikaci čtení RFID musí obsahovat NFC.

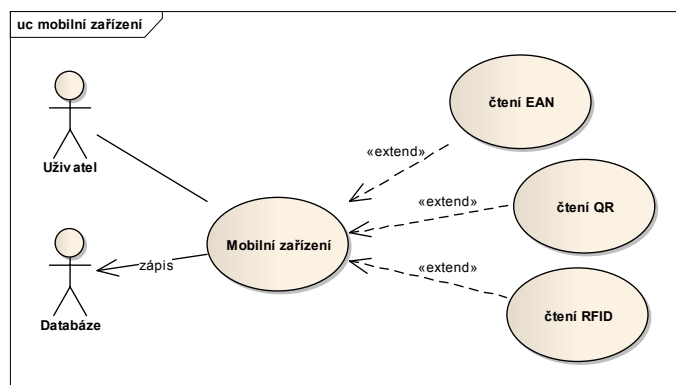


Diagram 9: Mobilní zařízení

### 8.2 Přínos databáze

- Databáze urychlí evidenci k pacientu
- Záznam pacienta bude okamžitě zpřístupněný více uživatelům
- Kontrola patientské péče
- Strukturalizovaný zápis (Rozvíjející se statistiky)
- Odpadá psaná forma ZIP, nahrazuje elektronický profil pacienta
- Možnost dalšího rozvoje

## 9 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat databázi pro urgentní příjem fakultní nemocnice Ostrava. Databáze měla sledovat moderní trend v IT, koncept *Internet Of Things*. Celý projekt byl rozdělen na tři práce. Načítání identifikačních kódů (EAN, QR, RFID), dále vytvoření databáze a vytvoření webového uživatelského rozhraní. Dohromady tato práce zahrnuje jak technické řešení práce, tak i týmovou práci.

Se získáním povolení výzkumu na urgentním příjmu, byla provedena ve spolupráci s personálem analýza, zaměřující se na polohu pacienta a práci s pacientem. Analýza zachycuje jen orientační práci a pohyb, jelikož zachytit veškeré úkony na urgentním příjmu bylo časově nesplnitelné. Výsledkem je přehledný obraz práce na urgentním příjmu, který je schválen interním zaměstnancem urgentního příjmu.

Práce byla implementována pomocí MySQL Workbench 5. 2. 42 CE. Toto uživatelské rozhraní umožnilo práci s databází, aniž bych použil dotazovacího jazyka SQL. Během práce jsem se seznámil i s jinými databázovými systémy, jako např. PostgreSQL, MSSQL, Firebird.

První navržený model databáze (ER diagram) byl nevhodný z hlediska uložení načtených identifikátorů. Musel být nahrazen robustnější databází rozšířenou o vazební tabulky. S tímto provedením se problém načítání identifikátorů vyřešil a databáze je funkční. Osobně si myslím, že by první model byl po menší úpravě taky plně funkční. Stávající model byl testován na rychlost odezvy dotazu. Byly vytvořeny dva dotazy po 99 opakování. Oba dotazy dosáhli přibližně 0,3 sekundy průměrného zpoždění.

Myslím si, že v blízké době nás automatizovaná evidence ať už ve zdravotnictví nebo jiných odvětvích nemine. S nástupem čipových občanských průkazů se evidence pacienta urychlí na maximum.

## Seznam použité literatury

- [1] KOCH, Miloš. Datové a funkční modelování. 2. vydání. Brno : Akademické nakladatelství Cerm, s.r.o, 2006. 108 s. ISBN 80-214-3252-7.
- [2] OPPEL, Andy. SQL bez předchozích znalostí : průvodce pro samouky. Brno : Computer press a.s., 2008. 240 s. ISBN 987-80-251-1707-1.
- [3] ŠARMANOVÁ, Jana. VŠB – TECHNICKÁ UNIVERZITA OSTRAVA. *Databázové a informační systémy: E-learningové prvky pro podporu výuky odborných a technických předmětů*. první. Ostrava, 2007. CZ.O4.01.3/3.2.15.2/0326. ISBN 978-80-248-1499-5. Dostupné z: <http://www.elearn.vsb.cz/archivcd/FEI/DAIS/DAIS.pdf>
- [4] FARANA, Radim. VŠB-TUO. *Databázové systémy: Microsoft Access 2.0*. 1995. vyd. Ostrava
- [5] MISHA. *Databáze: Architektura databází* [online]. první. 2010 [cit. 2013-05-06]. Dostupné z: <http://www.databaze.chytrak.cz/architektura.htm>
- [6] WIKIPEDIE.: Otevřená encyklopedie - Databáze. [online]. Poslední aktualizace 2009-11-17. Dostupné na internetu  
< <http://cs.wikipedia.org/wiki/Datab%C3%A1ze>>. [cit. 2013-01-05].
- [7] MILOŠ ŠEDA.: Databázové systémy [online]. Dostupné na internetu  
< [http://www.uai.fme.vutbr.cz/~mseda/DBS02\\_BS.pdf](http://www.uai.fme.vutbr.cz/~mseda/DBS02_BS.pdf)>. [cit. 2013-04-05]
- [8] WELLING, LUKE a Laura THOMSON. *MySQL: Průvodce základy databázového systému*. první. Brno: CP Books, a.s., 2005. ISBN 80-251-0671-3.
- [9] POKORNÝ, Jaroslav. Dotazovací jazyky. Praha: Karolinum, 2002. 255 s. ISBN 80-246-0497-3
- [10] ŘEPA, V. *Analýza a návrh informačních systémů*. 1. vyd. Praha: Ekopress, 1999. 403 s. ISBN 80-86119-13-0

## Seznam obrázků

Obrázek 1: Lineárního datového modelu .....	3
Obrázek 2: Hierarchický model dat .....	3
Obrázek 3: Síťový model dat .....	4
Obrázek 4: Objektový model dat .....	4
Obrázek 5: Relační model dat (Vlastní zpracování) .....	5
Obrázek 6: Centrální architektura .....	9
Obrázek 7: Architektura file-server .....	9
Obrázek 8: Architektura klient-server .....	10
Obrázek 9: Architektura distribuovaných databází .....	10
Obrázek 10 Popis ER diagramu .....	20
Obrázek 11 Obrázek 10: Typy relací. Plnou čarou se značí povinnost členství, čárkovanou nepovinnost. ....	20
Obrázek 12: Postup nového připojení k existující databázi .....	37
Obrázek 13: Schéma pro vytvoření tabulky .....	38
Obrázek 14: Modelování databáze ve Workbench .....	39
Obrázek 15: Ukázka toku dat .....	41

## Seznam tabulek

Tab. 1: Tabulka číselných datových typů v MySQL .....	15
Tab. 2: Řetězcové datové typy v MySQL .....	16
Tab. 3: Časové a datové typy v MySQL .....	16
Tab. 4: Ukázka datového slovníku .....	21
Tab. 5: Datový slovník charakterizující atributy v tabulce pacient .....	33
Tab. 6: Datový slovník entit vyhrazených pouze jako ZIP. ....	34

## Seznam diagramů

Diagram 1 .....	41
Diagram 2 .....	42

## Seznam příloh

Příloha A Datový slovník .....	I
Příloha B Povolení výzkumu .....	III
Příloha C ZIP .....	IV
Příloha D ER model .....	VI
Příloha E Diagram RFID .....	VII

## **Adresová struktura přiloženého DVD**

- Bakalářská práce
- Diagramy
- Přílohy
- Poznámkový blok



## Příloha A- Datový slovník

Typ Entity	Atribut	Datový typ	Délka	Klíč	NULL	IO	Poznámka
Vzorek	id	INT	100	A	N	Primární klíč	auto. Navýšení
	kod	VARCHAR	100	N	N		EAN, QR, RFID
	jmeno	VARCHAR	45	N	A		
	prijmeni	VARCHAR	45	N	A		
	Pacient_id	INT	100	A	N	Cizí klíč	
Zamestnanci	id	INT	100	A	N	Primární klíč	auto. Navýšení
	kod	VARCHAR	100	N	N		
	jmeno	VARCHAR	45	N	N		EAN, QR, RFID
	prijmeni	VARCHAR	45	N	N		
	telefon	INT	9	N	N		
	pozice	VARCHAR	100	N	N		
Revize_pristroju	id	INT	100	A	N	Primární klíč	auto. navýšení
	datum_revize	DATE		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	zamestnanci_id	INT	100	A	N	Cizí klíč	
Pristroje	id	INT	100	A	N	Primární klíč	auto. navýšení
	kod	VARCHAR	100	N	N		EAN, QR, RFID
	výrobní_cislo	VARCHAR	100	N	N		
	nazev	VARCHAR	50	N	N		
	lokalizace	VARCHAR	45	N	N		
Osobní_vecy	id	INT	100	A	N	Primární klíč	auto. navýšení
	cennosti	VARCHAR	100	N	A		
	místo_ulozeni	VARCHAR	45	N	A		
	datum	DATETIME		N	N		
	obleceni	VARCHAR	200	N	A		
	prirucni_zavazadlo	VARCHAR	45	N	A		
	kod_cennosti	VARCHAR	45	N	A		EAN, QR, RFID

\*Datový slovník vymezující entity nezasahující, nebo nepřímozasahující do ZIP.

Typ Entity	Atribut	Datový typ	Délka	Klíč	NULL	IO	Poznámka
Pacient_has_leky_ZIP	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Leky_ZIP_id	INT	100	A	N	Cizí klíč	
Pacient_has_Vykon_ZIP	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		

	Pacient_id	INT	100	A	N	Cizí klíč	
	Vykon_ZIP_id	INT	100	A	N	Cizí klíč	
Pacient_has_Konzilia_ZIP	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Konzilia_ZIP_id	INT	100	A	N	Cizí klíč	
Krevni_banku_ZIP_has_Pacient	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Krevni_banku_ZIP_id	INT	100	A	N	Cizí klíč	
Pacient_has_Vysetreni_ZIP	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Vysetreni_ZIP_id	INT	100	A	N	Cizí klíč	
Pacient_has_Pristroje	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	parametry	VARCHAR	200	N	A		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Pristroje_id	INT	100	A	N	Cizí klíč	
Pacient_has_Osobni_veci	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Osobni_veci_id	INT	100	A	N	Cizí klíč	
Konzilia_ZIP_has_Zamestnanci	id	INT	100	A	N	Primární klíč	auto. navýšení
	Konzilia_ZIP_id	INT	100	A	N	Cizí klíč	
	Zamestnanci_id	INT	100	A	N	Cizí klíč	
Laborator_ZIP_has_Vzorek	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Laborator_ZIP_id	INT	100	A	N	Cizí klíč	
	Vzorek_id	INT	100	A	N	Cizí klíč	

\*Datový slovník pro vazební tabulky

## Příloha B – Povolení výzkumu

VŠB-TU Ostrava  
Fakulta elektrotechniky a informatiky (FEI)  
17. listopadu 15/2172, Ostrava-Poruba, 708 33  
tel.: +420 596 919 353  
sekretariat.fei@vsb.cz

Fakultní nemocnice Ostrava  
17. listopadu 1790  
708 52 Ostrava-Poruba  
tel.: +420 597 371 111  
fno@fno.cz

### Žádost o povolení výzkumu


Já, **Tomáš Urbanczyk**, student 3. ročníku Vysoké školy báňské technické univerzity Ostrava oboru biomedicínský technik, **žádám o povolení k výzkumu na urgentním příjmu**, abych získal podklady k vypracování bakalářské práce s názvem: *Databáze pro evidenci označených entit sledující koncept Internet of Things*. Pod dohledem vedoucího práce Ing. Jakubem Jirkou.

\* Podklady se rozumí prázdné formuláře pro evidenci entit.

V Ostravě, dne 4. 4. 2013:



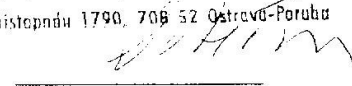
doc. RNDr. Jindřich Černohorský, CSc.  
VŠB-TU Ostrava, FEI  
Katedra kybernetiky a biomedicínského  
inženýrství




Tomáš Urbanczyk  
VŠB-TU Ostrava, FEI  
Biomedicínský technik  
3. ročník

FAKULTNÍ NEMOCNICE OSTRAVA  
Rc. Mária Dobešová

ředitelka oddělení pro ošetrovatelskou péči  
17. listopadu 1790, 708 52 Ostrava-Poruba

  
Fakultní nemocnice Ostrava

## Příloha C-Záznam intenzivní péče (první strana)



**FAKULTNÍ NEMOCNICE  
OSTRAVA**  
17. listopadu 1790, 708 52 Ostrava-Poruba

Místo pro nalepení štítku pacienta

### ZÁZNAM INTENZIVNÍ PÉČE

Pracoviště: Oddělení centrálního příjmu - URGENTNÍ PŘÍJEM

**Datum:**

**Čas příjmu pacienta:**

**Čas předání pacienta:**

**Místo předání:**

**Číslo:**

**Alergie:**

**Poznámka:**

**Výška:**

**Váha:**

**Cenné věci:** Trezor ☐ U sebe ☐ NE ☐

**Ošacení:** Šatna ☐ U sebe ☐ NE ☐

**Osobní věci:** ANO ☐ NE ☐

**Občanský průkaz:** ANO ☐ NE ☐

**Průkaz pojišťovny:** ANO ☐ NE ☐

**Časová osa vyšetření**

Interval měření		10					20					30					40					50				
Krevní tlak, pulz	240																									
	220																									
	200																									
	180																									
	160																									
	140																									
	120																									
	100																									
	80																									
	60																									
40																										
Infuze																										
Perfuzor																										
Ordinace																										
Ostání	TAT 0,5 ml i.m.																									
	TEGA 250 IU i.m.																									
	Stav vědomí																									
	Ventilace																									
	O <sub>2</sub> l/min																									
	SaO <sub>2</sub>																									
	VAS																									
	FLACC																									
	RAMSAY																									
	Odsávání																									
Tělesná teplota																										

\*Interní formulář fakultní nemocnice Ostrava (zdroj)

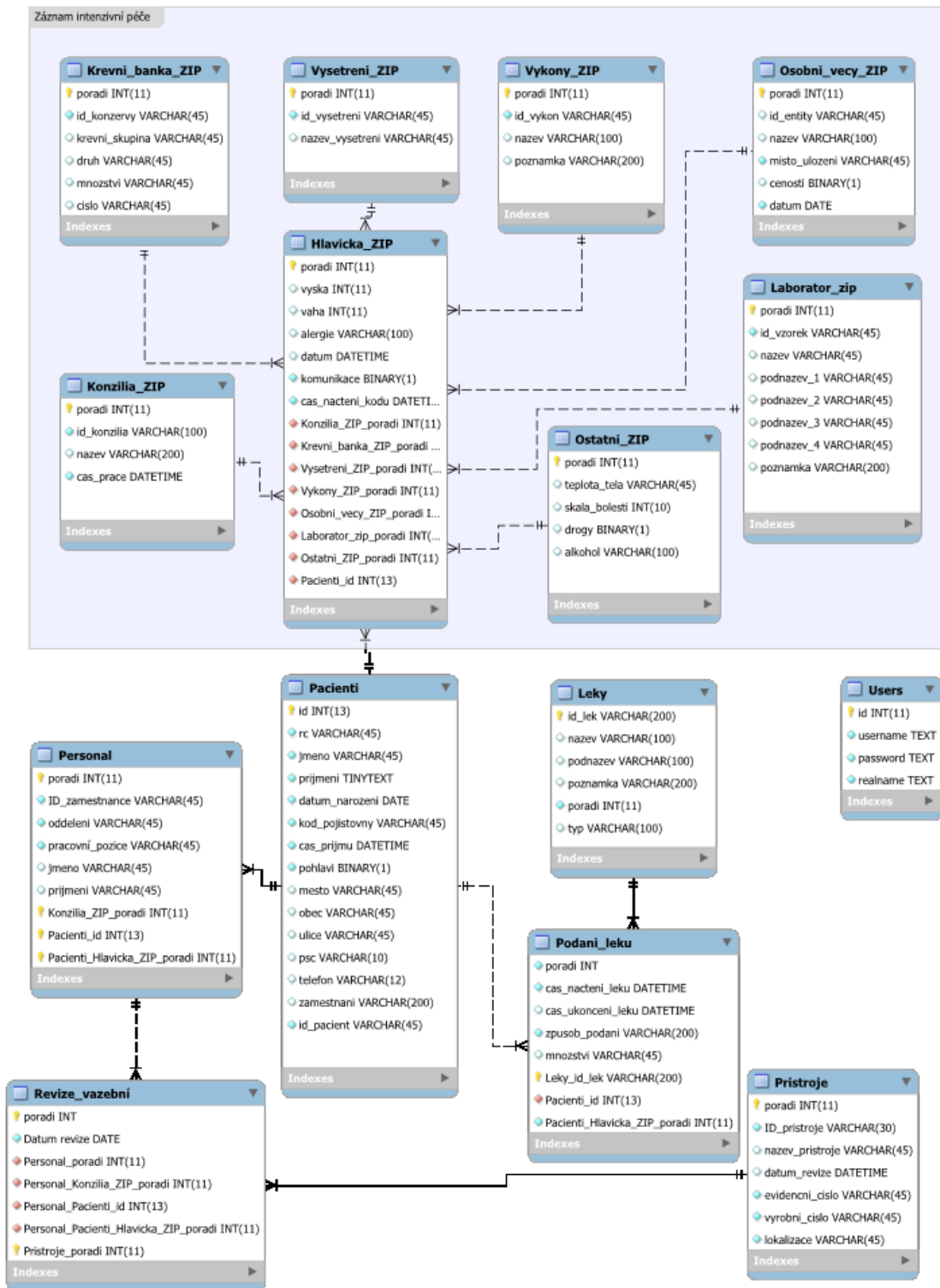


## Příloha C-Záznam intenzivní péče (druhá strana strana)

<b>LABORATOŘ:</b> <input type="checkbox"/> BIOCHEMIE <input type="checkbox"/> Základ (Na, K <sup>+</sup> , Cl, U, KREA, ALT, AST, GLY, OSM) <input type="checkbox"/> DN (Na, K <sup>+</sup> , Cl, U, KREA, ALT, AST, GLY, OSM, etanol, moč) <input type="checkbox"/> Kardiologie (Na, K <sup>+</sup> , Cl, U, KREA, ALT, AST, GLY, OSM, CK-MB, Tn) <input type="checkbox"/> Chirurgie (Na, K <sup>+</sup> , Cl, U, KREA, ALT, AST, GLY, OSM, ALP, BB, CRP, AMS) <input type="checkbox"/> KREVNÍ OBRAZ <input type="checkbox"/> KOAGULACE <input type="checkbox"/> BED SIDE <input type="checkbox"/> CRP <input type="checkbox"/> Moč <input type="checkbox"/> ..... <input type="checkbox"/> ..... Výsledky kritických hodnot: ..... ..... .....		<b>KREVNÍ BANKA:</b> KREVNÍ SKUPINA: <input type="checkbox"/> ANO <input type="checkbox"/> NE OBJEDNANÉ TP: ..... PODANÉ TP + KREVNÍ DERIVÁTY Druh / číslo / množství / čas podání <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> ..... <input type="checkbox"/> .....	
<b>ALKOHOL PRO SOUDNÍ ÚČELY:</b> <input type="checkbox"/> ANO <input type="checkbox"/> NE <b>TOXIKOLOG. VYŠETŘENÍ:</b> <input type="checkbox"/> Moč <input type="checkbox"/> Krev <input type="checkbox"/> Žaludeční obsah <input type="checkbox"/> ÚSL <input type="checkbox"/> Oddělení		<b>REGRES:</b> <input type="checkbox"/> ANO <input type="checkbox"/> NE	
<b>VYŠETŘENÍ:</b> <input type="checkbox"/> EKG <input type="checkbox"/> RTG <input type="checkbox"/> CT <input type="checkbox"/> CT+AG <input type="checkbox"/> MR <input type="checkbox"/> SONO <input type="checkbox"/> ECHO <input type="checkbox"/> FIBROGASTROSKOPIE <input type="checkbox"/> JINÉ..... <input type="checkbox"/> JINÉ..... <input type="checkbox"/> JINÉ.....	<b>VÝKONY:</b> Zavedeno: <input type="checkbox"/> PŽK..... <input type="checkbox"/> PŽK..... <input type="checkbox"/> CŽK..... <input type="checkbox"/> ART..... <input type="checkbox"/> OTI..... <input type="checkbox"/> PMK..... <input type="checkbox"/> NGS..... <input type="checkbox"/> VÝPLACH..... <input type="checkbox"/> HD..... <input type="checkbox"/> HD..... <input type="checkbox"/> SÁDRA..... <input type="checkbox"/> SUTURA..... <input type="checkbox"/> REPOZICE..... <input type="checkbox"/> EXTENZE..... <input type="checkbox"/> CELK. TOALETA..... <input type="checkbox"/> OHŘEV <input type="checkbox"/> CHLAZENÍ <input type="checkbox"/> JINÉ.....	<b>KONZÍLIA:</b> <input type="checkbox"/> ANESTEZIOLOG <input type="checkbox"/> ANESTEZIOLOG - DĚTSKÝ <input type="checkbox"/> TRAUMATOLOG <input type="checkbox"/> NEUROLOG <input type="checkbox"/> NEUROLOG - DĚTSKÝ <input type="checkbox"/> CHIRURG <input type="checkbox"/> PLASTICKÝ CHIRURG <input type="checkbox"/> NEUROCHIRURG <input type="checkbox"/> KARDIOLOG <input type="checkbox"/> INTERNISTA <input type="checkbox"/> ORL <input type="checkbox"/> STOMATOCHIRURG <input type="checkbox"/> OČNÍ <input type="checkbox"/> UROLOG <input type="checkbox"/> PSYCHIATR <input type="checkbox"/> ORTOPED <input type="checkbox"/> JINÉ.....	
<b>Poznámka:</b> ..... ..... .....			

\*Interní formulář fakultní nemocnice Ostrava (zdroj)

## Příloha D – Prvotní ER diagram



## Příloha E – Diagram práce s RFID

